

A Categorization Technique for Resolving Information System Failures Reasons

Sherif Mohamed Zahran¹ and Galal Hassan Galal-Edeen²

¹ Information System Department, Faculty of Information and Computers, Cairo University
PO box 12613, Orman, Giza, Egypt

Tel: 20-100-566-5543 E-mail: s.zahran@fci-cu.edu.eg

² Information System Department, Faculty of Information and Computers, Cairo University
PO box 12613, Orman, Giza, Egypt

Abstract— Detecting and resolving failures in information system (IS) projects is highly desirable process. Being able to do so depends on the proper identification of underlying reasons for failures; which are often difficult to accurately identify. One of the fundamental enablers of this identification is suitable categorization of the related reasons. This paper will concentrate on developing a categorization technique for IS projects failure reasons, identifying some of these categories, and providing a simple justification for reasons behind this failure. IS projects failure reasons categories are based on the underlying IS elements such as IS components, IS lifecycle, project environment, scope, budget, requirements and its technical and managerial issues. IS project may have failure reasons belong to more than one category at the same time and this may affect its importance and priority for resolving. This research covers most of IS failure reasons categorization to provide a valuable hierarchy to navigate through these reasons and their categories. This categorization is used in highlighting unknown, unexpected or unpredictable failures reasons during IS projects failure identification and analysis processes. The proposed technique of reasons categorization is experimentally used on a set of IS projects and its results are over expected. It discovers reasons that are extremely hidden and never expected to be directly or even indirectly related to IS failures.

Index Term— Context Modeling, Failure Analysis, Information System, Information architecture, Information Management, IS Failure Reasons Categorization, Project Management, Risk Analysis, Software Engineering.

1. INTRODUCTION

Billions of dollars are wasted on failed projects in developed countries every year. We don't expect that the situation in developing economies would be any better. Learning from failure can help these countries and developing countries preserve precious resources and avoid future failures of same nature, as well as avoid the opportunity cost of not using ICT where it would have been effective, because of the perceived risks of entering into new ICT projects. Heeks in [1] emphasized the fact that the study of failures can only take place post-hoc, once a failure has been identified. Failure handling requires special attention from information professionals. The process aims to reduce the number of system failures and

minimize their impact when they occur. The process involves diagnosing relevant system states, finding, analyzing, and addressing the reasons of failures, developing success criteria to avoid failures, and to recover when they occur. Any attempt to address the concept of success or failure in IS projects in developing countries, typically starts by categorizing success and failure [1]. Although, they may endure over time, though never perpetual, failure can be perpetual if proper actions are not taken. Information systems are particularly prone to failure. Some information systems never materialize; others appear late and/or over budget.

Consistent success requires sophisticated approaches which rely on good design practices and experience, so as to finally achieve desired goals that lead to the satisfaction of stakeholders, while avoiding undesirable outcomes [1]. This paper will address the categorization of IS failure reasons resulting in a four-level schema. The first level describes the category base, the second level highlights a sub categorization for the bases appeared in level one, the third level identifies IS failure reasons categories, and the last level depict a clear view of the details for the failure reasons. After classifying the failure reasons into categories, they will be prioritized according to the importance, urgency, required cost and available budget, and required time for recovery.

This paper will be organized as follows: section 2 gives basic definitions for failure in general and IS failures. Section 3 identifies the categories of IS failure and the categories of IS projects relative to success and failure. Section 4 overview the great negative effects of information system failures and the efforts made to minimize these negative effects. Section 5 frames the scope and limitations of this paper. Section 6 depicts the picture of IS failure reasons categories and proposed a new technique for IS Failure reasons categorization to help IS projects to recover from failure. Section 7 assures that flexibility of the categorization process by describing multi-categorization possibility. Section 8 applies the proposed technique on three case study projects to assure its validity and highlight its analysis results. Section 9 concludes this paper and outlines possible future work.

2. FAILURE DEFINITION

The Oxford English Dictionary mentioned that failure definition include the following: 1) lack of success, 2) the omission of expected or required action, 3) the action or state of not functioning, 4) an unsuccessful person or thing [2]. A context in which failure is frequently used is in the formal grading of scholastic achievement. 'Failing a test' or being assigned a 'failing mark' indicates that a student has submitted work or received a mark below a minimum threshold of performance or quality required to continue studies in a subject. Failure is defined in ISO/CD 10303-226 as the lack of ability of a component, equipment, sub system, or system to perform its intended function as designed. Failure may be the result of one or many faults[3]. Failure can be differentially perceived from the viewpoints of the evaluators. A person who is only interested in the final outcome of an activity would consider it to be an Outcome Failure if the core issue has not been resolved or a core need is not met. A failure can also be a process failure whereby although the activity is completed successfully, a person may still feel dissatisfied if the underlying process is perceived to be below expected standard or benchmark.

Jared Diamond lists some reasons by which a society can collapse: Failure to anticipate, or perceive a problem [4]. Failure analysis is the process of determining the reason of failure, collecting and analyzing data, and developing conclusions to eliminate the failure mechanism causing specific device or system failures. It is an important discipline in many branches of manufacturing industry, such as the electronics industry, where it is a vital tool used in the development of new products and for the improvement of existing products [5]. The concept of failure in the area of information systems however, is generally ill-defined [6]. Many researchers consider failure as self-evident requiring no further clarification [6], [5], [7].

An information system failure dispute arises from mismatched expectations between customer and developer, an ambiguously worded software contract, poor project management, or simply software that does not conform to specifications. A failed system or software project may cost millions causing undue cost in time and money and might lead to significant disruption to business operations[7]. These are some situations where Information Systems have been judged as failures: when the potential benefits of the system are not realized, when

the system is not used, or when there is substantial user resistance.

3. CATAGORIES OF FAILURE

Heeks in [1] categorizes the failure into two categories , **total failure** and **partial failure**. **Total failure** of an initiative never implemented or in which a new system is implemented but immediately abandoned. Such an outcome can be defined relatively objective. **Partial failure** of an initiative in which major goals are unattained or in which there are significant undesirable outcomes. In some cases, where only a sub-set of initially-stated objectives have been achieved, the notion of partial failure may be relatively straightforward [1]. A successful project must be on time, on budget, and deliver quality (features and functions) as promised. Anything less will be either a failed project or a challenged project [8].

The Standish Group categorized the information system projects in its chaos report produced in 1994 into the following three categories, **Project success** where a project is completed on-time and on-budget, with all features and functions as initially specified, **project challenged** where a project is completed and operational but over-budget, over the time estimate, and offers fewer features and functions than originally specified- , and finally **project impaired** where a project is canceled at some point during the development cycle [9].

The Standish group categorizes any project into (Project Success, Project challenged, Project impaired) whereas Heeks in [1] only categorizes failed projects into: total failure and partial failure. By definition, we can conclude that project challenged is in a partial failure and project impaired is in a total failure state.

4. IS FAILURE EFFECTS AND EFFORTS

The Standish Group announced in its chaos report produced in 1994 that: the success rate was only 16.2%, while challenged projects accounted for 52.7%, and impaired (cancelled) for 31.1%. [9]. But these percentages are in 2009 better than before as appearing in Table 1 [9–12]. The first step in resolving IS failures is to address the problem and define the failure reasons before any preservation or reflection actions. Any preservation system designers need a clear vision of the threats against which they are asked to protect content. Any preservation plan should address failure reasons like those are mentioned in Table 2.

TABLE I
The Standish Group Report, CHAOS from 1994 to 2009 Rates[9], [12][10], [13]

	1994	1996	1998	2000	2002	2004	2006	2009
Successful	16%	27%	26%	28%	34%	29%	35%	32%
Challenged	53%	33%	46%	49%	51%	53%	46%	44%

Failed 31% 40% 28% 23% 15% 18% 19% 24%

TABLE II
Generic IS failure reasons [14]

IS Failure Reason	Description
Media and Hardware Failures	Failure causes include random bit errors and recording track blemishes, breakdown of embedded electronic components, burnout, and misplaced off-line HDDs, DVDs, and tapes.
Software Failures	All practical software has design and implementation bugs that might distort communicated data.
Communication Channel Errors	Failures include detected errors (IP packet error probability of ~10 ⁻⁷) and undetected errors (at a bit rate of ~10 ⁻¹⁰), and also network deliveries that do not complete within a specified time interval.
Network Service Failures	Accessibility to information might be lost from failures in name resolution, misplaced directories, and administrative lapses.
Component Obsolescence	Before media and hardware components fail they might become incompatible with other system components, possibly within a decade of being introduced. Software might fail because of format obsolescence which prevents information decoding and rendering within a decade.
Operator Errors	Operator actions in handling any system component might introduce irrecoverable errors, particularly at times of stress during execution of system recovery tasks.
Natural Disasters	Floods, fires, and earthquakes.
External Attacks	Deliberate information destruction or corruption by network attacks, terrorism, or war.
Internal Attacks	Misfeasance by employees and other insiders for fraud, revenge, or malicious amusement.
Economic and Organization Failures	A repository institution might become unable to afford the running costs of repositories, or might vanish entirely, perhaps through bankruptcy, or mission change so that preserved information suddenly is of no value to the previous custodian.

5. SCOPE AND LIMITATIONS

This paper mainly gathers, identifies and highlights most of IS failure reasons then categorizes them into a set of categorizations to provide a valuable hierarchy to navigate through these reasons and their categories. The covered failure reasons and their categories belongs to the popular IS types like, Transaction Processing System (TPS), Management Information System (MIS), Decision Support Systems, Execution Information Systems (EIS), Expert Systems (ES), Communication and collaboration systems (CCS) and Office automation systems (OAS) [15]. The advanced, complex information systems like, Geographic Information System (GIS) and General Dynamics are not covered in this paper. The proposed technique for resolving IS failure reasons is structured, tested and applied on a set of real cases in the governmental and private sectors. This technique categorizes IS failure reasons into a four-level hierarchy.

Then, it surveys these categories to check if there are any reasons causing failure in the context of the underlying IS in each case study. The effectiveness of the proposed technique mainly focuses on discovering, organizing and describing hidden and unexpected reasons of IS failures and recommending solutions for each reason.

The main issue that will be an excellent complementary for this technique is how to prioritize the discovered reasons

before resolving since the time dimension and the order of resolving steps are extremely critical in the process of IS failure recovery.

6. ANALYSIS OF FAILURE FOR INFORMATION SYSTEM FAILURE

Unfortunately, many projects, particularly those involving information systems, fail to deliver against their objectives on time and within budget [1]. Most information systems fail either totally or partially [7]. IT projects are almost going over budget and schedule, always missing expectations and their investment return is very poor [7]. Charette defined some factors to be the most common factors of software failure, like: unrealistic or unarticulated project goals, inaccurate estimates of needed resources, badly defined system requirements, poor reporting of the project's status, unmanaged risks, poor communication among customers, developers, and users, use of immature technology, inability to handle the project's complexity, sloppy development practices, poor project management, stakeholder politics, and commercial pressures [16]. Bronte-Stewart in [7] mentioned that IT projects failure seems to highlight the same problems and probable reasons of failure. He provides an comparison for 5 studies of IT projects failure reasons which are depicted, summarized and compared in an empirical way[7]. He compares failure reasons from the following 5 reports: Research reports by a Government select committee (2000), OASIG (1996),

National Audit Office (NAO)/Office of Government Commerce (OGC) (2002), Standish Group CHAOS (1995) [9] and Schmidt et al (2001) [17] that identify typical reasons of failure/success in IT projects [7].

In following subsections, IS failure reasons will be categorized based on certain categorization bases that we propose to help identifying, grouping, categorizing, neither neglecting nor forgetting any failure reason. This proposed categorization may also provide a significant participation in IT success, recovering from failure and avoiding falling in failures. This research identifies eight categorization bases which cover most of IS project components, lifecycle, environment, scope, budget, technical and managerial aspects. These bases are described in the following subsections. These subsections justify each base, and define the categories inside each one of them. In each subsection, each categorization base will be highlighted by a figure of its own categories and examples from existing failed information systems caused by this category reasons..

6.1 Component-Based Categorization

O'Brien, in [15] defines IS as “it can be any organized combination of people, hardware, software, communications networks, and data resources that collect, transforms, and disseminate information in an organization” [15]. A system that accepts data resources as input and processes them into information products as output; a system that uses the resources of hardware, software and people to perform input, processing, output, storage and control activities that transform data resources into information products; a purposefully designed system that brings data, computers, procedures, and people.

From the above definition of IS, We proposed the first scheme for categorizing IS failures reasons which is based on IS component, namely: hardware, software, communications networks, people, data resources and organization. A sample case for this categorization in a health information system failed by the software category of reasons is as follows: “Problems began to occur with the implementation in 1999. These can be divided into problems related to infrastructure, application, and organization of the implementation process” [18]. Because there were too many proposed functions to implement in one phase, some hospitals ended up trying to run the information system in a reduced form in parallel with separate pharmacy and laboratory systems. Many of the modules that were proposed in the initial plan were not created in time. The software also used advanced features, such as replication, that staff then had to be trained in, causing delays [18]. Figure 1 highlights the first categorization base of IS failure reasons which is the component-based categorization. The key observations from a study mentioned in [19] are: (1) system software

and hardware failures are the two major contributors to the total system downtime (22% and 10%) [19]. As mentioned, the main failure reasons are all around IS components, mainly software application.

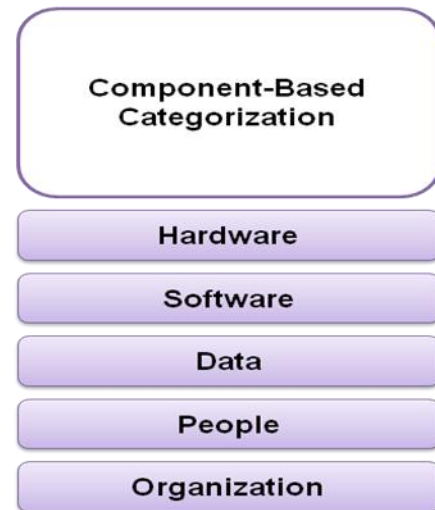


Fig. 1. Component-Based Categorization

6.2 Lifecycle -Based Categorization

Software development came to be viewed as a large activity involving efforts to establish requirements, design a solution, develop or implement the designed solution, test to verify, and validate the solution to ensure it correctly, and fully addresses the established requirements. It may also be described as a long term process that does not end by the product delivery, but rather extends the product lifecycle in terms of update, and maintenance that have to be done by a large number of people from different backgrounds all working towards the same target [20], this is namely IS lifecycle. Information system could fail at any stage of its project development, starting by its initial conception through to its implementation and subsequent maintenance. So, analyzing IS failure can be helpful to consider these consequent stages. One common description of the stages of an information system project is known as the systems development lifecycle (SDLC). Although there are many variants, the SDLC has the following basic structure: feasibility study, systems investigation, systems analysis, systems design, implementation, review and maintenance [21]. Not every project will require that the phases be sequentially executed. It depends upon system development methodology the project following. However, these phases are interdependent. Depending upon the size and complexity of the project, phases may be combined or overlapped according to the selected SDLC Model.

This categorization is based on SDLC phases where each phase is considered a separate IS failure category. Thus, the categories will be: System analysis, System design,

System implementation, System testing, System installation and customization, and any other phases as required and operated for the underlying project. [19] observed that maintenance and configuration contribute to 24% of system downtime [19]. Interestingly enough the survey reveals the major reasons of project failure during the lifecycle of the project are a breakdown in communications (57%), a lack of planning (39%) and poor quality control (35%) [22]. Figure 2 categorizes failure reasons according to SDLC phases. It mentions the main phases and aggregates the other phases in one category as an assumption since the categorization hierarchy is designed to be as much generic and flexible as possible.

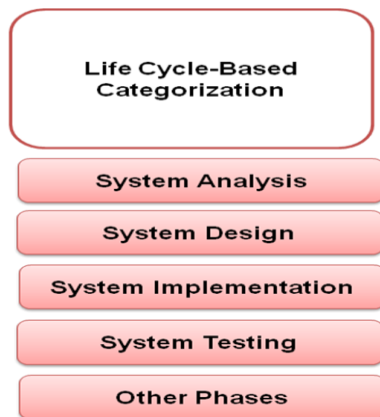


Fig. 2. Lifecycle-Based Categorization

6.3 Environment-Based Categorization

IS may be judged against different set of standards than those originally envisaged. In other words, the environment may have changed so that IS is now expected to do things that were not envisaged when the original system was designed [21]. So, the environment change may be an important reason for IS failure. Whether something is judged to be part of the environment is determined by whether or not it influences or is influenced by the system that has been perceived. The environment can also exert a degree of control over the system, but while the environment can be influenced by the system it cannot be controlled by the system [21]. Hwang et al in [23] proposed a success model for Management Information System (MIS) depending on six variables as appeared in figure 3, five of them are environmental variables. This is a simple indicator about the importance of the environment in the success or failure of MIS or any Information system.



Fig. 3. A system Success Model [23]

These three basic concepts of system, boundary and environment are remarkably powerful tools for reaching an understanding of or explaining failures in information and other systems [21]. Figure 4 divides the environment categorization according to information system boundary and its external environment to inside project and outside project. Furthermore, the outside project category is divided depending on project organization and project country.



Fig. 4. Environment-Based Categorization

6.4 Technical-Based Categorization

The success or failure of a system cannot totally be seen as dependent on the technical perspective - other considerations are vital to the success of the development and implementation of a system [24]. Yet another categorization approach is based on technical or non-technical tasks which come from the technical-based categories. It should be a general categorization that covers all the technical materials, phases, stages, machines, devices and their related failure reasons. Also, another category of all the non-technical issues should be separated from the above mentioned failure reasons. This categorization can be considered as a mandatory categorization for IS failure reasons since it covers all the

related aspects of any information system. Also, it intersects with all the other categorizations, or we can say, it covers them all.

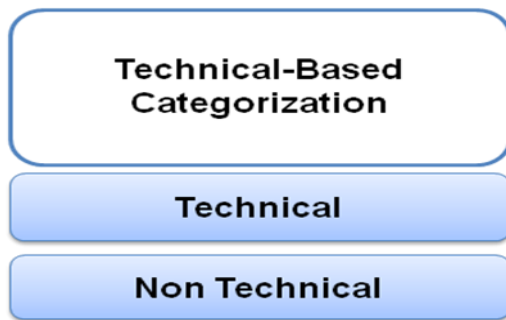


Fig. 5. Technical-Based Categorization

With every passing day, software is becoming more and more complex. With each new line of code, new bugs and security risks introduced in software. The software complexity is currently an obligatory factor which aggravate when unsafe programming languages are used, that do not protect against simple kind of attacks, such as buffer overflows [25]. This is a good example for technical causes. Also, Technical category of IS failure causes may contain: Poor development practices, Poor design modeling, Careless quality control, Inaccurate installation and customization, Missing developed component. Non-technical category of IS failure causes may contain: vague requirements, poor fit between systems and organization, inadequate user training, lack of user satisfaction. The critical role played by social and organizational factors in the implementation of IS in developing and developed countries has been noted many times and its effects and consequences are really significant [1], [26]. Figure 5 gives a high level view for these categorizations, dividing it into technical and non-technical categories. Each category has its own details and subcategories, some of them are mentioned above.

6.5 Scope-Based Categorization

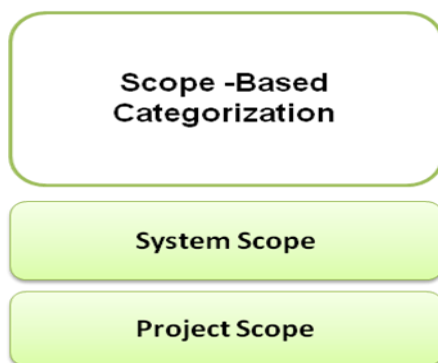


Fig. 6. Scope-Based Categorization

Defining system scope and project scope will identify the most important characteristics of any Information system. Some of the most important reasons for IS failure are related to system scope and project scope as described in figure 6. Thus, we can categorize failure reasons as they relate to system scope which encompasses external organization and project scope which encompasses internal components and functionality. There are many IS failure reasons in system scope category such as: cultural mismatches or program customization whereas, reasons inside project scope category are like: ill defined scope, uncovered functionality or unorganized components. Projects fail too often because the project scope was not fully appreciated and/or user needs not fully understood [8] and 31.1% not fulfilled the scope [27].

6.6 Budget/Size -Based Categorization

organizations and governments spend an estimated \$1 trillion on IT hardware, software, and services worldwide. Of the IT projects that are initiated, from 5 to 15 percent will be abandoned before or shortly after delivery as hopelessly inadequate. Many others will arrive late and over budget or require massive reworking. Few IT projects, in other words, truly succeed [16]. Several catastrophic failures in large scale real time systems can be attributed to the inadequacy of existing interfaces and the inability to track implicit assumptions of components [28], [29]. A project's sheer size is a fountainhead of failure. Studies indicate that large-scale projects fail three to five times more often than small ones. The larger the project, the more complexity there is in both its static elements (the discrete pieces of software, hardware, and so on) and its dynamic elements (the couplings and interactions among hardware, software, and users; connections to other systems; and so on). Greater complexity increases the possibility of errors, because no one really understands all the interacting parts of the whole or has the ability to test them [16].

It is well known that the majority of information systems run drastically over-budget or fail altogether. Various studies have found that between 40% and 60% of ISD projects fails to meet budget estimates and that the degree of overspend can exceed 200% [30]. Although somewhat dated, the Standish Group conducted one of the most extensive and often cited studies which showed that, of 8,000 projects, only 16% were completed within budget [9]. Furthermore, it should not be assumed that tightening budgetary control is always a good thing; while tight budgetary control is often positively co-related to budgetary performance, this is not always the case, and overly restrictive budgetary control can stifle a project and increase the chance of failure [30]. As mentioned in figure 8, small projects are more likely to succeed than large

projects [12]. Figure 7 divides failure reasons categories as it is divided in Standish Chaos report, 1999 [12].

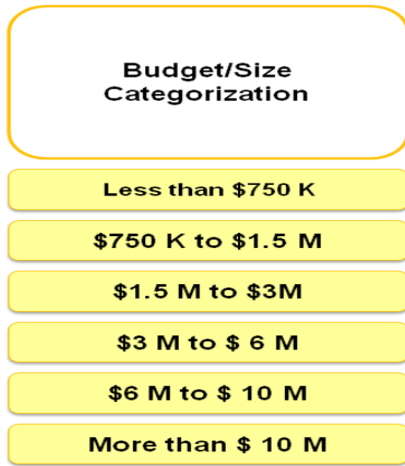


Fig. 7. Budget-Based Categorization

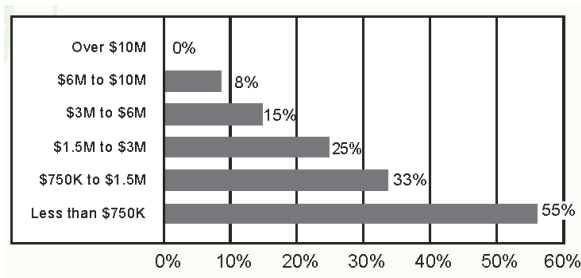


Fig. 8. Success by Project Size[12]

6.7 Requirements-Based Categorization

Just as the specifications of user requirements are vital in the stages of design and computer system development, so can the knowledge and regulations which constitute the basis for SDLC model selection determine the success or failure of a given project [31]. Since user requirements are the core of any information system, any requirements misunderstanding or unclearness may cause a severe failure. User requirements are often unclear at the start and can change at any stage in the project’s lifecycle [7]. There are functional requirements, non-functional requirements and system requirements. System requirements may include the organization and the external environment requirements which contain the cultural constraints. Lack of project user satisfaction is one of the popular reasons for failure [32]. As it appears in figure 9, these three types of requirements form one on the most critical proposed categorization in this paper which is the requirements-based categorization.

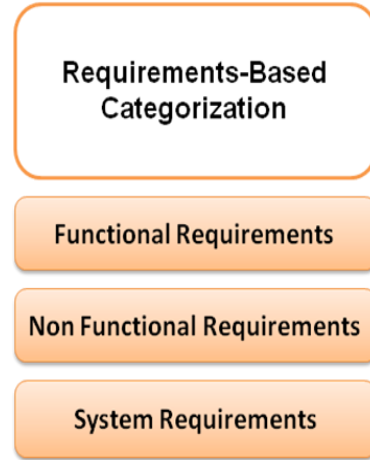


Fig. 9. Requirement-Based Categorization

Requirements engineering is relatively general process. It can be considered as an early step in the process of software engineering. It does not have an exact or common definition but there are some definitions that clarify its concepts, contents and steps. Requirements engineering is defined in accordance to [24] as "a systematic process of developing requirements through an iterative co-operative process of analyzing the problem, documenting the resulting observations in a variety of representation formats, and checking the accuracy of the understanding gained." [20]. Functional and non-functional requirements are the other two categories inside the requirements-based categorization. The combination of the functional and non-functional requirements formulates the user requirements which are the source of the user satisfaction. User satisfaction is the first success criterion for any IS project [32].

Failures in information systems are not only technical in nature. They are often caused by lack of attention to social, cultural and ethical factors. This problem is exacerbated by the emergence of the new economy and the consequent rise in the number of trans-national information systems. Cultural differences may give rise to differing perceptions of ethical issues and differing approaches to ethical reasoning [24]. Since Information systems are social activity systems, they should not be regarded as an only technically-driven artifact, but also social and human factors driven phenomena. So, the culture of system environment affects the information system success and failure enablers dramatically. Different environmental culture in the developing countries from developed countries may helps some projects to succeed or at the same time, it may increase the other projects failure and never succeed.

6.8 Managerial-Based Categorization

Justifying the success/failure of software projects is a long standing problem, and managers for the past decades have expressed concerns about the value they are getting from software projects, and they have been searching for ways to evaluate and justify the use of software projects [33]. Software project management is a series of activities associated with carrying out the project as effectively as possible. The program of crisis management is not just a collection of mechanical actions or procedures, rules and the efforts of the mentality and the other it is a series of steps and processes to assess the mental well thought out and deal with the crisis, the real size.

Projects fail mainly because of unable to plan and estimate correctly, or fail to implement the tasks according to plan or failure causes by human factor. There are three reasons of failure [11], [27]. First, planning and Estimation factor; refers to initial cost and schedule estimates are not revised when more information becomes available as a project progresses. Also plans are not used correctly or used to guide the project forward, thus causing the project to fail. Second, implementation factor; caused by project scope changes, incorrect use of project methodology, major changes in the requirements and testing, and/or inspections are poorly done. Third, human factor; project managers are not trained to acquire the necessary management skills. Also, some managers are not able to apply and put the theory of project management into practice. Poor communications are also one of the human factors that cause a project to fail [27], [34]. Figure 10 depicts failure reasons managerial categorization. It is internally spitted into three levels: top, middle and low level management. They are mentioned without any identification or further clarification which depends on projects type and organization structure and complexity.

7. IS FAILURE REASON MULTI-CATAGORY

IS failure can be caused by a reason from only one category of the above mentioned categories which are accumulated in figure 11. It also may involve a reason or more from two, three or maybe more categories at the same time. These involved categories can be from the same categorization base or form another one. A good example of this case is the embedded systems when they fail. While it is possible to design embedded systems as separate hardware and software systems, failure to consider tradeoffs between choice of hardware engine and

design of application software can lead to designs that are too expensive, too slow, and never perform as intended. A key element of our understanding of co-design is the study of system modeling in all its forms [35]. This information system failure involves hardware, software, budget and time categories of failure reasons.



Fig. 10. Managerial-Based Categorization

8. CASE STUDY ANALYSIS

The above mentioned technique for IS failure reasons categorization is applied on three case study projects: first project is a TPS for property Ownership department in a governmental agency. The second one is an ERP for an import and export company. The third project is a social media monitoring project for private sector organizations. Table 3 highlights how the categorization technique affects the process of failures reasons exploration and its high impact to move IS projects from a failure state to a success state. It mentioned the result of failure reasons exploration process before using categorization technique and after using it.

The black cell means that there are no failure reasons in this category. The (×) means that this category reasons are not discovered. The (√) means that this category reasons are successfully discovered.

These results appearing in the table describes how the proposed technique affects the process of discovering IS failure reasons greatly. The proposed technique helps in moving projects from a failure state to a success state and clarifies reasons that are hardly or maybe impossibly to be discovered.

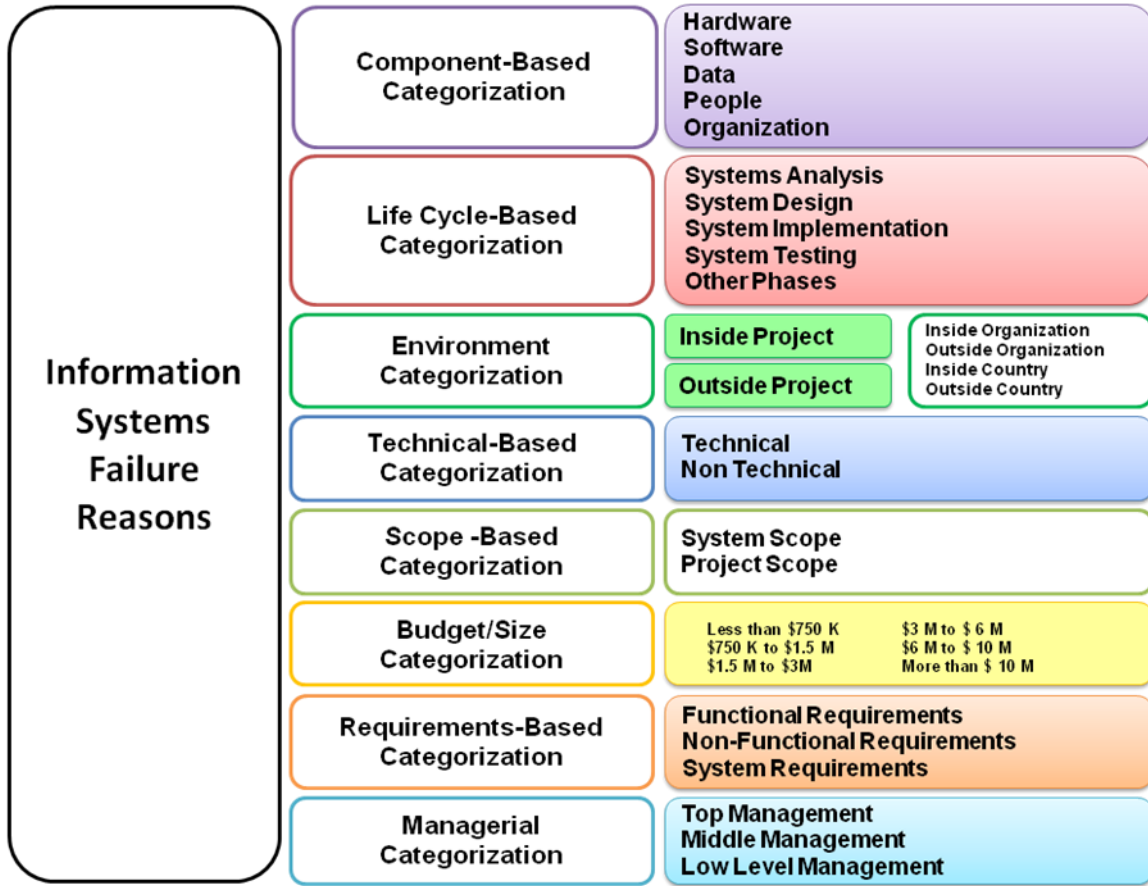


Fig. 11. Proposed IS Failure Reasons categories

9. CONCLUSION AND FUTURE WORK

This paper provides a categorization technique for IS projects failure reasons and gives an overview of IS failure reasons. The proposed technique helps in identifying all the materialized failures and their reasons, organizes the view of IS failure reasons, minimizes the required time and effort to get rid of these failures and facilitates detecting more failures and their reasons. It provides a comprehensive and wide range covering of failure reasons. It enables researches, Project managers, analysts to see a much big and clear picture of projects failure symptoms and reasons to be ready and able to recover sooner, cheaper and more accurate. The technique is applied on three case study projects in different context even physically in different countries and produces unexpected positive results. The case study projects prove that the proposed categorization technique helps greatly in

detecting IS failures reasons that are hidden, unexpected or appears to be not relative.

Estimating the effectiveness and the efficiency of these proposed categorization bases of IS projects failure reasons is a highly recommended work that could be applied through more case studies. Our research utilizes a qualitative research approach to characterize information systems failures. It starts with IS failure categorization, as discussed in this paper, then a prioritization technique will be built and applied on case studies since it will be an excellent complementary for the categorization technique by precisely determine how to prioritize the discovered reasons before resolving since the time dimension and the order of resolving steps are extremely critical in the process of IS failure recovery. Qualitative research techniques, like grounded theory, will be investigated and studied to explain and utilize their empirical effects.

TABLE III
Case Study Projects Before and After Applying Categorization Technique

<i>IS Failure</i>	<i>IS Failure</i>	<i>Project One</i>	<i>Project Two</i>	<i>Project Three</i>
-------------------	-------------------	--------------------	--------------------	----------------------

<i>Categorization</i>	<i>Category</i>	<i>Before</i>	<i>After</i>	<i>Before</i>	<i>After</i>	<i>Before</i>	<i>After</i>
<i>Component</i>	<i>Hardware</i>	x	✓	x	✓	x	✓
	<i>Software</i>	✓	✓	x	✓	✓	✓
	<i>Data</i>	✓	✓	✓	✓	x	✓
	<i>People</i>	x	✓	x	✓	x	✓
	<i>Organization</i>	x	✓	x	✓	x	✓
<i>Lifecycle</i>	<i>Analysis</i>	x	✓	x	✓	x	✓
	<i>Design</i>	x	✓	x	✓	x	✓
	<i>Implementation</i>			x	✓	x	✓
	<i>Testing</i>			x	✓	✓	✓
	<i>Other Phases</i>	x	✓	x	✓	x	✓
<i>Environment</i>	<i>Inside Project</i>	✓	✓	✓	✓	✓	✓
	<i>Outside Project</i>	x	✓	x	✓	x	✓
<i>Technical</i>	<i>Technical</i>	✓	✓	x	✓	✓	✓
	<i>Non-technical</i>	x	✓	✓	✓	x	✓
<i>Scope</i>	<i>Project Scope</i>	✓	✓	✓	✓	x	✓
	<i>System Scope</i>	x	✓	x	✓	x	✓
<i>Budget/size</i>	<i>Less than\$750K</i>	x	✓	✓	✓	x	✓
	<i>\$750K to \$1.5M</i>	x	✓	x	✓	x	✓
<i>Requirements</i>	<i>Functional</i>	✓	✓	✓	✓	✓	✓
	<i>Non-functional</i>	x	✓	x	✓	✓	✓
	<i>System Req.</i>	x	✓	x	✓		
<i>Managerial</i>	<i>Top Management</i>	x	✓	✓	✓	x	✓
	<i>Middle Management</i>	✓	✓	x	✓	✓	✓
	<i>Low level Management</i>	x	✓	x	✓	x	✓
Total		≈25%	≈100%	≈30%	≈90%	≈30%	≈95%
Status		Succeeded		Abandoned		Challenging	

REFERENCES

[1] R. Heeks, "Information Systems and Developing Countries: Failure, Success, and Local Improvisations," *The Information Society*, vol. 18, no. 2, pp. 101–112, Mar. 2002.

[2] M. J. Turpin, "Redefining failure: Phenomenology and meaning," *Australian occupational therapy journal*, vol. 55, no. 4, pp. 283–6, Dec. 2008.

[3] S. M. Pike, "Distributed Resource Allocation with Scalable Crash Containment," The Ohio State University, 2004.

[4] J. Diamond, *Collapse: How societies choose to fail or succeed*. Penguin Group, 2005.

[5] P. L. Martin, *Electronic failure analysis handbook: techniques and applications for electronic and electrical packages, components, and assemblies*. McGraw-Hill Professional, 1999.

[6] A. Kaniclides and C. Kimble, "A framework for the development and use of executive information systems," *Proceedings of GRONICS*, vol. 2, no. 2, pp. 47–52, 1995.

[7] M. Bronte-Stewart, "Developing a risk estimation model from IT project failure research," *Computing and Information Systems*, vol. 9, no. 3, pp. 8–31, 2005.

[8] R. Frese and V. Sauter, "Project success and failure: what is success, what is failure, and how can you improve your odds for success," *University of Missouri – ST. Louis*, 2003. [Online]. Available: <http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:PROJECT+SUCCESS+AND+FAILURE+WHAT+IS+SUCCE>SS

[9] The Standish Group, "The standish group report ©, Chaos," 1995.

[10] The Standish Group, "The Standish Group Report, CHAOS Summary 2009: The 10 Laws of CHAOS," The Standish Group, 2009.

[11] The Standish Group, "Trends Report, Trends in IT Value," 2008.

[12] The Standish Group, "The Standish Group Report, CHAOS: A Recipe for Success," 1999.

[13] The Standish Group, "The Standish Group Report, Extreme Chaos," 2001.

[14] H. M. Gladney, *Preserving digital information*. Springer, 2007.

[15] J. O'Brien, *Introduction to Information Systems*, 15th Revis. Mcgraw Hill Higher Education, 2009, p. 608.

[16] R. Charette, "Why software fails," *IEEE spectrum*, vol. 42, no. 9, p. 36, 2005.

[17] M. Keil, P. E. Cule, K. Lyytinen, and R. C. Schmidt, "A framework for identifying software project risks," *Communications of the ACM*, vol. 41, no. 11, pp. 76–83, 1998.

[18] P. Littlejohns, J. Wyatt, and L. Garvican, "Evaluating computerised health information systems: hard lessons still to be learnt," *Bmj*, vol. 326, pp. 860–863, 2003.

[19] J. Xu, Z. Kalbarczyk, and Ravishankar K. Iyer, "Networked Windows NT system field failure data analysis," in *Proceedings of the 1999 Pacific Rim International Symposium on Dependable Computing*, 1999, pp. 178–185.

[20] A. Mahdy, M. Ahmed, and S. M. Zahran, "Flexible CASE Tools for Requirements Engineering: A benchmarking Survey," in *International Conference of Informatics and Systems*, 2008, no. 2, pp. 20–26.

- [21] J. Fortune and G. Peters, *Information systems: Achieving success by avoiding failure*. Wiley: Chichester., 2005.
- [22] S. J. NetoAlvarez, "PROJECT MANAGEMENT FAILURE: MAIN CAUSES," Bowie State University, 2003.
- [23] M. I. Hwang, J. C. Windsor, and A. Pryor, "Building a knowledge base for MIS research: A meta- analysis of a systems success model," *Resources Management Journal*, vol. 13, no. 2, pp. 26–32, 2000.
- [24] D. Oram and M. Headon, "Avoiding information systems failure: culturally determined ethical approaches and their practical application in the new economy.," in *International scientific conference*, 2002, pp. 9–13.
- [25] M. A. Mehmood, A. S. Khokhar, and M. Ali, "Incorporating Security in Embedded System – A critical analysis," *International Journal of Computer Science Issues*, vol. 8, no. 6, pp. 156–160, 2011.
- [26] E. J. Hartnett, E. M. Daniel, and R. Holti, "Client and consultant engagement in public sector IS projects," *International Journal of Information Management*, vol. 32, no. 4, pp. 307–317, May 2012.
- [27] I. Attarzadeh, "Project Management Practices : The Criteria for Success or Failure," *Communications of the IBIMA*, vol. 1, pp. 234–241, 2008.
- [28] Ratneshwer and A. Tripathi, "Dependence Analysis of Component Based Software through Assumptions," *International Journal of Computer Science Issues*, vol. 8, no. 4, pp. 149–159, 2011.
- [29] A. Tirumala, T. Crenshaw, L. Sha, G. Baliga, S. Kowshik, C. Robinson, and W. Witthawaskul, "Prevention of failures due to assumptions made by software components in real-time systems," *ACM SIGBED Review - Special issue: The second workshop on high performance, fault adaptive, large scale embedded real-time systems (FALSE-II)*, vol. 2, no. 3, pp. 36–39, Jul. 2005.
- [30] K. Conboy, "Project Failure en Mass: A Study of Loose Budgetary Control in ISD Projects," *Sprouts: Working Papers on Information Systems*, vol. 8, no. 40, 2008.
- [31] A. B. Sasankar and V. Chavan, "SWOT Analysis of Software Development Process Models," *International Journal of Computer Science Issues*, vol. 8, no. 5, pp. 390–399, 2011.
- [32] I. Elpez and D. Fink, "Information systems success in the public sector: Stakeholders' perspectives and emerging alignment model," *Issues in Informing Science & Information Technology*, vol. 3, pp. 219–232, 2006.
- [33] M. Tarawneh, H. Tarawneh, and F. Alzboun, "A model for Crises Management in Software Projects," *International Journal of Computer Science Issues*, vol. 8, no. 6, pp. 112–116, 2011.
- [34] P. W. G. Morris, "Why projects fail," *Coley Consulting*, 1999. [Online]. Available: <http://link.aip.org/link/IEESEM/v1999/i29/p1/s1&Agg=doi>.
- [35] W. H. WOLF, "HARDWARE-SOFTWARE CO-DESIGN OF EMBEDDED SYSTEMS," *Citeseer*, vol. 8, no. 9402008, 1994.