IS/IT Troubled Projects: Searching for the Root-Causes

Paolo Rocchi
IBM Italia, Roma, Italy
LUISS University, Roma, Italy
procchi@luiss.it
Stefano Za
LUISS University, Roma, Italy
sza@luiss.it

Abstract

Purpose: Since decades troubled IS/IT projects continue to lose billions of dollars all over the world. The present paper looks for the root-cause of this broad noteworthy phenomenon.

Method: This paper begins with a concise survey of empirical research which shows how a significant part of project failures derives from the managers' behavior rather than from objective-technical obstacles. Researchers have made significant efforts to improve the professional conduct of managers however the situation has enhanced at small rate. This discrepancy raises doubts about a possible hidden cause lying behind the perceived determinants of failure, and the present paper applies the techniques typical of 'root-cause analysis' (RCA).

Findings: IS/IT managers handling troubled projects worldwide have nothing in common except for the same cultural basis. When students, they got fundamental concepts of informatics, therefore, it is natural to hypothesize a precise cause/effect relation between the lessons learned by the managers and their ineffective behaviors. A close analysis shows how computing theories are narrow, self-referential and abstract; they negatively prepossess the conduct of managers who have broad and practical responsibilities.

Research Implications: The cultural causes of runaway projects require to advance theoretical research about the foundations of computer science, so that a comprehensive and consistent construct will meet the multifold duties of IS/IT project leaders. The future theoretical edifice will underpin innovative courses in universities and professional organizations which will make the leaders more aware of their roles and capable of managing them.

Originality/Value: Speaking in general, when one traces back a problem to its origin, the one can eradicate the problem. This work brings evidence that the root causes of troubled projects reside in the culture of managers and therefore the professional preparation of managers needs to be improved so to remove the present difficulties in IS/IT projects.

Keywords — Cultural factors of troubled IS/IT projects; root cause analysis; theoretical computer science; managerial and operational responsibilities.

1 Introduction

The literature defines the successful project as that meets its core objectives in terms of quality, cost, time and scopes (PMI 2013). Frequently the projects which set up information systems with the support of information technology (IS/IT) do not match with this definition. IS/IT plans of work often exceed the tolerance levels set out by stakeholders and managers. Sometimes the customers do not derive the benefits for which the project was initiated. Quite a number of statistical surveys describe this puzzling phenomenon which began to emerge in the 1970s (Glass 2006)(Cerpa and Verner 2009).

The majority of surveys converges on the conclusion that successful IS/IT projects constitute a fraction while much of them are in a troublesome state (Sarif et al 2018)(Harwardt 2018). The Standish Group (2016) that periodically publishes the Chaos Reports shows how only around 30 percent of programs meet all the targets; nearly 50 percent are challenged notably they were completed but late, over budget or with unsatisfactory results. Approximately 20 percent of projects fail that is to say they were canceled or not used, in particular, large and grand plans of works fail to attain the intended targets (Table 1). The figures yearly fluctuate, but even when the success rates become somewhat higher, a significant percentage of works fail or are troubled. In substance, organizations private and public as well continue to waste millions of dollars due to this inexplicable phenomenon.

Some raised doubts about the published surveys (Thomas and Fernández 2008)(Eveleens and Verhoef 2010), hence Basten and colleagues (2013) made a global study and have demonstrated how the inquiries dealing with the IS/IT projects offer rather consistent views.

Statistical works have inventoried a broad assortment of determinants of project faults (Liu et al 2012) which they tend to gather into three major groups, and from (Nelson 2008) we get the following summary:

Process errors: The majority of fault determinants are mistakes made by managers in relation to processes (≈ 45%) e.g. Insufficient planning, lack of resources, changing requirements and specifications.

People errors: This group includes defects of managers in relation to people (≈ 43%) e.g. Unrealistic expectations, lack of user involvement, lack of IT management, ineffective stakeholder management.

Mixed errors: The remaining factors derive from a variety of causes not belonging to the previous: technical, legal, operational etc.

TABLE 1: Performances by Project Size (Standish Group 2016)

	Successful	**Challenged**	**Failed**
Grand	2%	7%	17%
Large	6%	17%	24%
Medium	9%	26%	31%
Moderate	21%	32%	17%
Small	62%	16%	11%
Total	100%	100%	100%

The largest groups of errors prove by no means that the IS/IT projects are impaired and ultimately cancelled because the managers are unable to organize the processes and to have effective relations with stakeholders, customers, users, technicians etc. (Haggerty 2000), while objective-technical factors have trivial influence. Various professional figures – e.g. IT managers, project leaders, senior managers, executives etc. – are not up to the expectations, and certified professionals do not perform managerial activities better than others (Catanio et al 2013).

In consequence of this situation perduring since decades we have conducted an articulated inquiry. The first stage recalls the research driven to amend the project managers' behaviors (Section 2) and the modest benefits obtained induced us to search for the root cause of the persistent errors (Section 3). The root-analysis methods have thought us how to hypothesize precise cultural-technical factors which have been verified from the theoretical viewpoint in Section 4, and from the professional practice viewpoint in Section 5. Section 6 draws the implications of the present findings.

2 Attacks from various directions

The determinants of project failures cited in the Introduction have prompted a large ensemble of inquiries. Researchers across different areas have addressed the issues with the scope of improving the professional abilities of IT managers. They have attacked challenging topics on different fronts. We confine ourselves to concise notes just to give an idea of the numerous explored areas and the multifold research programs undertaken to eliminate IS/IT project failures.

Computer experts started with confronting technical difficulties and at successive stages put forward the methods for software development labeled: Structured, Waterfall, Prototyping, Object-Oriented and Agile just to mention the most popular (Whitten et al 2003).

Sociologists and psychologists have analysed and tackled the so called 'people errors'. Some look into the psychological motivations of IS/IT executives (Rodriguez et al 2017); others notice how a leader should possess effective decision-making abilities, leadership and guidance skills in order to reach the desired objectives (Lee et al 2014). Rob Kling (1980) claims that the behavior of project leaders is a consequence of computer science (CS) courses that do not include the social perspective. Martinsons and Chong (1999) hold that proactive and supportive human relations yield greater user satisfaction, smoother organizational change and improved productivity, but do not significantly affect perceived output quality.

Experts of management science have scaled to the 'process errors' and teach how to optimize the modes of planning, organizing and delineating responsibility for the completion of the intended goals and how to refine organizational politics (Kerzner 2009)(Laudon and Laudon 2009). Bushuyeva and colleagues (2018) go through the competence development of an IT/IS team using the model of the International Project Management Association (IPMA) as a reference. Some scholars argue even about the legal implications of projects (Dhillon 2002) and the ethical implications (Reynolds 2018).

The International Organization for Standardization (ISO) together with the International Electrotechnical Commission (IEC) has delivered various publications presently superseded by (ISO/IEC 25010) that categorizes eight software quality attributes. ISO/IEC analyse, among the others, the functional suitability, the performance efficiency, the usability, the security and the maintainability of the software products, and explain how to measure them from the user or external perspective and from the technical or internal position. Documents assess the software development processes and the related business functions; they write down how to acquire information for users, how to arrange the software life cycle process etc.

The mentioned references are just a fraction of the exceptionally large amount of inquiries, lessons, recommendations, case studies, and surveys conducted by the scientific community. The overall literature appears to be impressive, however the most recent data cited in the Introduction prove that the results miss the requisite qualities. It is surprising that the IS/IT project issues are not over after such *numerous and multifaced initiatives,* and the behavior of managers has not ameliorated in a significative manner after decades.

This conclusion persuaded us to search the profound reasons of the persevering project faults.

3 Root-cause analysis

Problematic IS/IT projects present traits that are unique and do not emerge in other engineering sectors.

3.1 Natural doubts

It may be that the building of a house falls behind schedule, a car can have mechanical defects, a chemical plant pollutes the environment etc. Troubled projects occur in all the engineering sectors however the rates of failures, especially of the catastrophic ones, appear to be much lower respect to software engineering. The International Project Leadership Academy (IPLA 2019) yearly surveys big works achieved in transport, retail, building etc. and shows how the works involved with the software technology make a significant group.

As second, managers very rarely make mistakes like those of IT leaders. For instance, the determinants labeled as 'insufficient planning' and 'unrealistic expectations' seldom occur elsewhere (Maher and Hadidi 2017). Most factors leading to uncontrolled projects might be classified as 'elementary mistakes' and give the impression that the IS/IT managers ignore the basic organizational criteria and rules.

As third, from the end of the Second World War we are witnessing the greatest progress in science and technology ever made in the history of mankind. This astonishing phenomenon attracted the attention of various scholars such as Firestein (2016) who explores the reasons of science' success. The massive increase of computing power, cheap memory and large data storage have sustained the brilliant advance of the digital technology which contrasts with the IS/IT projects absorbing energies and moneys like black holes. The high percentage of challenged and failed programs counterpoints the smart progress of IT, and the rudimentary mistakes of managers have inspire new reflections (Dahlberg and Kivijärvi 2016)(Streit and Pizka 2011); they even lead to the central question of the present research paper:

Why the scientific community has not won the match against troubled projects?

3.2 From the effects back to the causes

The statistical surveys played a significant role so far since they have supplied indications to cut the causes of faults. The numerous determinants of failures – for instance, Nelson (2008) censuses 36 different failure factors – have guided the multifold research approaches in a way. The details vary within the inquiries, the perspectives, and the scopes of reviewers so the spectrum of failure factors have widened to a certain extent. The attempts to neutralize the defects identified by statistical studies proved to be correct beyond any doubt, yet they might be insufficient.

When an incident occurs, management science teaches to consider whether there is something more important that needs attention. It may be that behind the perceived and lower-level factors, a higher-level and less-evident factor sets in motion the entire cause-and-effect reaction which ultimately leads to the problem under examination. The expert who traces back a problem to its origin, can eradicate the problem, therefore, the present inquiry has the purpose of seeking for the initiating causes of IS/IT project failures.

The 'root cause analysis' (RCA) is the systematic and pragmatic process used in industry, business, medicine etc. for searching and correcting the most significant causes of crucial events (Okes 2019). Effective management requires more than watching the facial symptoms of a problem and RCA seeks for the out-of-sight reasons. Hence, the first goal of RCA is to discover the root cause of a problem. The second goal finds a remedy and corrects the intended problem. Factual experience proves that the solutions addressing the root cause drop the obstacles in a definitive manner. Eventually, the third objective of RCA applies the learned lesson to prevent future obstacles or to obtain further positive results.

3.3 Looking from above

The conclusion of Section 2 and the remarks of Sub-section 3.1 deal with broad phenomena, they look at the issues from above. Instead of focusing on a few special cases, this research examines the question globally and supervises the runaway projects occurring worldwide. RCA opens diverse ways to reach the three goals (Doggett 2006), but this inquiry faces the IS/IT projects globally and it is sufficient to study closely a pair of measures:

i) - The inspection of (partially) failed solutions that should have prevented the undesired consequences.

ii) - A systematic canvassing of the effects and the connected causes.

Sections 1 and 2 deals with point i), that is the missing victory of scientists against problematic projects. It remains to explore point ii) by means of interrogative techniques.

The root-cause analysis puts forward various standards such as the Five Whys, the Fish Bone and the Gemba Gembutsu method (Latino and Latino 2006). Here we follow the ensuing route to articulate the origin of troubled projects occurring world-wide.

The literature registers difficulties which are not confined to a few teams, to a kind of organization, or to a special institution. The inventoried troubled projects occur in the five continents, in advanced and developing economies alike, in any sector of activity, in public and private entities etc. Moreover, the faults repeat since fifty years at least. Therefore, the process and people errors cited in the Introduction raise the ensuing precise queries:

A) *What does negatively influence managers who use different languages, who are living inside different habitat and cultures, and have worked in distant organizations since decades? What have they in common?*

The answer is not complicated. Computer experts who are working in the five continents have nothing in common except for the same technical education; not to say that they have attended the same lessons, or followed the same courses, or adopted the same didactical material, simply they share the same conceptual basis on computing. Note how they did not learn identical technical notions which are rapidly evolving, instead all of them studied the fundamentals of CS including the Turing machine, the recursive theory, the Shannon entropy and so forth. The core topics are universal for all the students because informatics is a mathematical-engineering discipline grounded in precise theories that did not vary with time. Even the managers who do not have technical extraction apprehended the theoretical basis somehow. Numerous IS/IT projects are run by lawyers, economists and humanists, who as prior condition attended lessons of CS and thereby shared the same cultural heritage.

Concluding, it seems very reasonable to answer questions **A** *this way:*

B) *IS/IT leaders have a sole common experience: the acquisition of the foundational notions of computer science at school; thus the scarce professionalism of IS/IT managers and in turn the project faults have a cultural origin, they derive from the basics of CS.*

This profound cause of distressed projects has been hypothesized through the deductive reasoning outlined in Italics and the reader probably asks:
What bibliographical and practical evidence does support this diagnosis?

The state of the art of computer science needs proper verification, and we shall check the profile of computing as discipline. In the second stage we shall control the impact of factors **B** on the professional practice.

4 The profile of informatics?

Since the 1950s the immature state of CS appeared controversial and raised significant criticism.

4.1 The uncertain core of computer science

Donald Knuth (1973) wrote: "Having surveyed the relationships of computer science with other disciplines, it remains to answer the basic questions: What is the central core of the subject? What is it that distinguishes it from the separate subjects with which it is related?".

Tedre (2015) shows how the relationships between computing and mathematics raised debates since the earliest age. Forsythe (1967) and others aimed to go beyond this interpretation and wrote: "The ultimate purpose of physics is the intellectual one of understanding the physical world [...] Similarly we may expect that some day the agreed ultimate purpose of computer science will be to understand the behavior of information and the laws which govern its processing." Edsger Dijkstra (1986) looked into the contrasting perspectives emerging from theoretical studies and system engineering. Peter J. Denning (1989) has given contribution to the maturation of computer science as a discipline and is going on to advocate the definition of general principles (Denning and Martell 2015). Other thinkers worked in favor of the progress of CS, but the clear cultural structure of CS still lacks. The queries of Knuth remain unanswered in the substance, and the controversial essence of informatics supports conclusion **B** blaming the insufficient preparation of IS/IT managers.

4.2 Theoretical computer science

If this cultural remonstration is felt as a generic argument by the reader, it remains to go deep into *theoretical computer science* (TCS) that gathers all the constructions about computing (Arbib et al 2012) and for this reason gives the basics of CS and the subject contents to the university lessons of managers to be. The editorial (Rocchi 2016a) presents a detailed analysis of TCS; this paper merely remembers two cultural features typical of TCS: *narrowness* and *self-referentiality* which have a precise relation to IS/IT leaders' questionable professionalism.

Speaking in general, a theory has the scope of explaining and forecasting events occurring in the world. The broader is the theory, the more events it can illustrate and the more services it can offer. On the other side, a narrow construct proves to be of little utility both for academicians and practitioners.
In spite of this essential requisite, each computing construct focuses on a narrow area. The very names of theories show how each one explores a limited field. E.g. There is no general computational theory, five constructions tackle specialist computational issues. E.g. The queuing theory focuses on lining up elements that will be processed. E.g. Relational algebra serves to model a relational database and define queries on it. In summary, each part of TCS refers to a particular topic and turns out to be somewhat trivial from the intellectual viewpoint.

The history of science makes us informed about researchers who discovered partial results at subsequent stages. By time passing, they merged those results and made a unified consistent domain of knowledge. Even when scientists started with wrong ideas, they progressively changed direction and created one admirable intellectual edifice. The partial views of TCS do not constitute an unsurmountable defect since various inquiries have supplied material to correct and enrich the initial frame. For instance, researchers laid down the *laws of software evolution* (Kour and Singh 2016), the *software quality* and the *software metrics*. *Semioticians* fixed the general properties of signs and information (Krampen et al 1987).

However, computing theorists do not build up any conceptual web, they leave out these and other contributions. What is worst, they do not harmonize the components of TCS, and do not integrate even the constructions that treat the same topic. E.g. The theory of coding, the Shannon theory and cryptology deal with discrete signals and should undergo integration instead they present disjoint results. E.g. Every software program is made by instructions and data, but the theories of algorithms and data structures lie apart one from the other. Because of this separation, informatics consists of several notions that do not have any logical-formal link and do not offer a unified and exhaustive view of the field.

In conclusion, TCS consists of fragmentary parts. The grand theory which should articulate the core of computer science is missing and corroborates the authors just cited who blamed the cultural defeat of CS, in turn all this matches with the conclusion **B**.

Let us examine how TCS prepossesses the professional behaviors of IS/IT leaders.

5 Job responsibilities

The digital revolution is an international enterprise, sustained by thousands and thousands of experts who share the concepts of computing put on trial by **B** and nonetheless bring about extraordinary solutions. Several young wizards excogitate wonderful products and create myriads of apps. Experts make design of any complexity in robotics, space flights and other advanced fields. It is natural to object:

Why the lessons learned at school result only in the scarce professionalism of IS/IT managers?

5.1 Broad and narrow responsibilities

An IS/IT leader directs the works and has a broad responsibility, this means that he fulfils a broad assortment of duties. He analyses the client's demand; he evaluates the available resources; he prepares the budgets and negotiates the contracts. He interacts regularly with the stakeholders, the suppliers and the collaborators. He looks into the alternative potential solutions. He has concern with the success of the artifact and also has to prevent adverse effects to people, the environment etc. In summary, a leader needs to get the complete vision of the artifact to prepare and even to master the subordinate aspects dealing with the laws, economics, accounting and psychology. This just to show how the fragmentary notions of informatics oppose the mission of executives who need the 360° view of the work that they are undertaking. The partial theories of TCS sustain the statement **B** since the dramatically counteract the job of project managers who have to master multifold topics.

By contrast, a practitioner who works in a specific field, employs a narrow construction of TCS without problems, in fact he works in a precise field and has limited responsibility. E.g. Network engineers use the queueing theory as an effective tool to assess the peaks of access to the Internet. The narrow coverage of the queueing theory does not prove to be a defect for them. We even recall the managers who have limited duties when the project is small, or the project is large but sets up a purely technical artifact that exclusively involves engineering issues. It may happen that a leader means to offer a ready product to the market and does not need to negotiate or customize the solution. The range of his responsibility narrows and the cultural deficiency of TCS does not have influence.

5.2 Acquired notions and managerial performances

Even if programming is no longer perceived as the soul of informatics, programming still towers as the central theme for theorists who usually claim *computer science is a branch of mathematics.* Alan Turing, a cornerstone in TCS, integrates this idea and holds that *a software program is the solution to a mathematical problem*. Several lessons learned at school spawn from these premises and result in a sequel of distorted professional attitudes. Robert Glass (2000) holds: "Academics, I would assert, are largely in the 'math is vital' camp. Practitioners are in the 'math is of little consequence' camp."

- ✓ The managers, when young, considered important the themes emphasized by computing theorists even if they might have negligible value in the professional practice. For example, commonly undergraduates go deep into the algorithm *complexity*, and learn the *Big O notation* used to qualify computer programs, whereas everyday experience shows how these topics have secondary impact respect to human factors (Bieri 1955).

- ✓ The pupils used to confront programming through the techniques typical of those who reason in abstract. The students apprehended to address an algorithm as an intellectual challenge, like a race of intelligence. Future managers started on this path and began to assign minor value to the software development methods which teach how to implement a software application in plain manner. The management of people, resources, money and other material entities went in the background that instead has significant weight (Coombs 2003). The future leaders prepared for *computational thinking*, were not fully aware how their hands would get dirty in the living environment as Denning (2017) notes.

- ✓ Abstract lessons introduced a scenario populated with mathematical algorithms, instead the professional practice presents a different landscape. For instance, Wikipedia (2019) makes the complete list of free and open source software packages offered to consumers. The list groups the programs into 32 categories and only two categories (out of thirty-two) treat mathematics and statistics. Not only software programming does not coincide with mathematics, but the latter applications make a minority group in the modern world.

- ✓ A software package is a product to implement and to sell, thus the ISO/IEC documents detail how it must comply with several criteria for quality. Computing theories underscore the ‘correctness’ of programs (Streicher 1991) and place behind the remaining more relevant quality factors for users and customers such as usability, efficiency, flexibility etc.

- ✓ If the software program is the solution to a mathematical problem as Bates and Constable (1985) argue then it should be in service without interruption. The correct solution to a mathematical problem remains forever and never changes; instead, the contrary occurs in the working environment. Maintenance turns out to be a relevant process in software management (April and Abran 2012) which TCS disregards. It is natural for working leaders to underestimate the software maintenance which many consider as an accident and not as an important job commitment. Witte (2017) highlights this reductive view: “Vendor maintenance fees are often based on a fraction (usually 12 percent to 18 percent) of current list price. (...) As a result, maintenance costs are underestimated by a factor of four or more.”

- ✓ The literature associates IS/IT projects to *software engineering*, yet a clear difference emerges between traditional engineering and software engineering. The first designs or anyway controls its solutions by calculations. Any artifact must conform with precise numerical parameters. By contrast, universal experience shows how software metrics does not play a leading role in IS/IT projects. Software products are neither designed nor confirmed by equations and a manager cannot sustain or explain a solution using undisputable numerical values. The missing measures can cause the disorientation and unsatisfaction of customers and stakeholders. Once the chief executive officer of a company said to me: “I’ve made a great investment in computer hardware and software, but I’m unaware whether I purchased a lorry, a car or a motor-cycle”.

- ✓ A leader cannot be successful without well-developed communication skills (Monteiro 2014); he must be able to clearly communicate tasks, goals and expectations. The project manager of any engineering domain is normally able to illustrate all the elements of the future artifact in a simple manner. His scientific background allows him to explain the details of the design using plain expressions. For example, a civil engineer describes the building to erect with the support of diagrams, maps and univocal terms. By contrast, an IS/IT manager has taken lessons underpinned by incomplete and sometimes contradictory discourses. He sees the project consisting of heterogeneous and partially antagonistic concepts. The narrow, disjoined and abstract parts of TCS do not aid the IT designer to make clear the features of the future information system to the customers and stakeholders. Hashim and colleagues (2013) make a summary of various surveys dealing with IT project failures and highlight that ‘poor communication’ is the risk factor cited in 13 surveys over 20 reports.

- ✓ The technical jargon of project managers has a broad and rather confusing assortment of definitions in consequence of theoretical fragmentation. For example, many words are in use for a single item and a single term can denote different and incomparable items (read numerous examples in (Laplante 2001)). In practice, TCS does not teach univocal concepts and, what is worst, TCS offers a terminology that compounds human relations. One reasonably suspects that some *people factors* – e.g. ‘Lack of user involvement’ and ‘Ineffective stakeholder management’ – are not due to the asocial personal tendency of managers. It is impossible that all the IS/IT managers in the world have a disposition hostile to social practices. It is more logical to conclude that confusing terminology provoke confusing messages.
  The lack of univocal terms becomes a terrible flaw since IS/IT managers – as the executives of whatsoever engineering field – must communicate with different people. Long since researchers have made several attempts to make uniform the language of IT experts but the studies of taxonomy (Dwivedi et al 2013) have obtained moderate benefits.

5.3 Approximate cultural recover

Probably, the reader objects that a working manager updates his professional knowledge on the basis of job training, courses, vocational education, books etc. (Steshenko et al 2017). Even the most disorganized school does not impose a life sentence, and students improve their personal preparation with time.

This is undoubtedly true, but when a manager adjusts his ideas, he does not have the authority of academic professors. He advances by trials and errors as long as a precise intellectual reference does not exist. He tentatively grasps new ideas while keeps the old contradictory concepts. He continues to recognize a level of excellence to ethereal notions since TCS still echoes in his mind. His uncertain culture calls to mind an individual who engages in a project as a pastime rather than as a profession. The low percentages of successful projects and the high rate of challenged and failed works remind us the performances of amateurs rather than well formed professionals.

6 Implications from the research

The present work casts light on a broad landscape and leads to develop a ponderous research strategy in theoretical computer science so that the preparation of managers meets the expectations. Scientists should go beyond the current fragmentary and dated frames, and should justify the more recent technical achievements too. They should design a grand theory of computing explaining the multiple facets of technology. In substance, this paper relaunches the appeals of Dijkstra, Knuth, Denning, Freeman, Hart and others in order to found computer science on general principles. Those authors were convinced that IT has progressed enough that it is now possible to describe its intellectual substance in a new and exhaustive way. They argued in favor of informatics as a technology-oriented discipline whose formalized fundamentals underpin all the IS applications. They advocated the science of information processing which interacts with the world, notably it should relate to business, industry, society etc. in effective manner and eventually will unravel the enduring problems of IS/IT leaders. When informatics will become a mature discipline – au par with electronics, chemistry and others – managerial issues under discussion will be tackled and solved all over the world.

7 Conclusion

IS/IT projects driven to provide customized solutions manifest in over budget, poor-quality, unmanageable and overdue upshots. Numerous surveys show how considerable part of failures originates from the conduct of managers. Researchers coming from various fields have put forward a number of suggestions in order to remedy managers' lack of professionalism, although, after decades the performances of managers have progressed so slowly as troubled projects remain very significant and highly costly for business.

We decided to search whether a concealed obstacle hinders experts and found out that the ineffective behaviors of IS/IT leaders have cultural origin. The literature presents various reflections about the cultural impact on projects (Schein 2010) (Alvesson 2013) and also on IT projects (Gu et al 2014), but they mostly focus on the *organizational culture*. Thinkers argue on leadership, adhocracy, competing values, business, corporate identity and so forth. Instead, the present work points the finger at the insufficient *technical culture* of managers. It has proved how leaders share fractioned, partial and abstract notions about the software technology; it has demonstrated how the cultural preparation inspires ineffective behaviors and in turn frustrates the efforts of researchers who mean to help managers but brought about a moderate advance.

This neglected intellectual causes of runaway projects imply to progress in the theoretical domain and to set up a unifying exhaustive frame. Thinkers and scientists should double their efforts to improve TCS. In the close it is worth mentioning the personal contribution to reach a comprehensive and integrated views of the essentials of computing (Rocchi 2013)(Rocchi 2016b), a plan of work still in progress.

References

Alvesson M. (2013) - Understanding Organizational Culture - (2nd Edition) SAGE Publications.

April A., Abran A. (2012) - *Software Maintenance Management: Evaluation and Continuous Improvement* - J. Wiley & Sons.

Arbib M.A., Kfoury A.J., Moll R.N. (2012) - *A Basis for Theoretical Computer Science* - Springer.

Basten D., Pankratz O., Joosten D. (2013) - Assessing the assessors: An overview and evaluation of IT project success reports - *Proc. of the 21st European Conf. on Information Systems*, pp. 1-12.

Bates J.L., Constable R.L. (1985) - Proofs as programs - *ACM Trans. Program. Lang. Syst.*, Vol. 7, No. 1, pp. 113-136.

Bieri, J. (1955) - Cognitive complexity-simplicity and predictive behavior - *Journal of Abnormal and Social Psychology*, Vol. 51, No. 2, pp. 263-268.

Bushuyeva N., Bushuiev D., Busuieva V., Achkasov I. (2018) - IT projects management driving by competence - *IEEE Intl. Scientific and Technical Conference on Computer Sciences and Information Technologies* (CSIT), Vol. 2, pp. 226-229.

Catanio J.T., Armstrong G.,Tucker J. (2013) - The effects of project management certification on the triple constraint - *Intl. J. of Information Technology Project Management*, Vol. 4, No. 4, pp. 93-111.

Cerpa N., Verner J.M. (2009) - Why did your project fail? - *Comm. of the ACM*, Vol. 52, No. 12, pp. 130-134.

Coombs P. (2003) - *IT Project Estimation: A Practical Guide to the Costing of Software* - Cambridge University Press.

Dahlberg T., Kivijärvi H. (2016) - Towards an integrative, multilevel theory for managing the direct and indirect impacts of it project success factors - *Proc. 49th Hawaii Intl. Conf. on System Sciences*, pp. 4971-4980.

Denning P.J., Comer D.E., Gries D., Mulder M.C., Tucker A., Turner A.J., et al. (1989) - Computing as a discipline – *Computer*, Vol. 22, No. 2, pp. 63-70

Denning P., Martell C.H. (2015) - *Great Principles of Computing* - MIT Press.

Denning P.J. (2017) - Remaining trouble spots with computational thinking - *Comm. of the ACM*, Vol. 60, No. 6, pp. 33-39

Dhillon G. (ed) (2002) - *Social Responsibility in the Information Age: Issues and Controversies* - IRM Press

Dijkstra E.W. (1986) - On a cultural gap - *The Mathematical Intelligencer*, Vol. 8, No. 1, pp. 48-52.

Doggett M. (2006) - Root cause analysis: A framework for tool selection - *Quality Management Journal*, Vol. 12, No 4, pp. 34-45.
Dwivedi Y., Ravichandran K., Williams M., Miller S., Lal B., Antony G., Kartik M. (2013) - IS/IT project failures: a review of the extant literature for deriving a taxonomy of failure factors - *Proc. Intl. Conf. on Transfer and Diffusion of IT*, pp.73-88.
Eveleens J.L., Verhoef C. (2010) - The rise and fall of the Chaos Report figures - *IEEE Software*, Vol. 27, No.1, pp. 30-36.
Forsythe G.E. (1967) - A university's educational program in computer science - *Comm of the ACM*, Vol. 10, No. 1, pp. 3-11.
Glass R.L. (2000) - New answer to 'How important is mathematics to the software practitioner?' - *IEEE Software*, Vol. 17, No 6, pp. 135-136.
Glass R. (2006) - Looking into the challenges of complex IT projects - *Comm. of ACM*, Vol. 49, No. 11, pp. 15-17.
Gu V., Hoffman J., Cao R., Schniederjans M. (2014) - The effects of organizational culture and environmental pressures on IT project performance: A moderation perspective - *Intl. Journal of Project Management*, Vol. 32, N. 7, pp. 1170-1181.
Haggerty N. (2000) - Understanding the link between IT project manager skills and project success research in progress - *Proc. of the ACM Conf. on Computer Personnel Research*, pp. 192-195.
Harwardt M. (2018) - Project success from the management perspective: A quantitative evaluation - *Open Journal of Information Systems*, Vol. 5, No. 1, pp. 24-51.
Hashim R., Abbas M., Hashim M. (2013) - Critical success factors assessment in software projects - *Proc. of Conference on Science and Information*, pp. 282-287.
IPLA (2019) - *Catalogue of Catastrophe, International Project Leadership Academy*. Available at: http://calleam.com/WTPF/?page_id=3#2015
ISO/IEC 25010 - *Systems and Software Engineering - Systems and Software Quality Requirements and Evaluation — System and Software Quality Models.* Available at: https://webstore.iec.ch/preview/info_isoiec25010%7Bed1.0%7Den.pdf
Kerzner H. (2009) - *Project Management: A Systems Approach to Planning, Scheduling and Controlling* - 10th edition, John Wiley & Sons Inc.
Kling R. (1980) - Social analyses of computing: Theoretical perspectives in recent empirical research - *Computing Surveys*, Vol. 12, No. 1, pp. 61-111.
Knuth D. (1973) - *The Art of Computer Programming* - 2nd edition, Addison Wesley Longman Publishing Co.G.
Kour G., Singh P. (2016) - Using Lehman's laws to validate the software evolution of agile projects - *Proc. Intl. Conf. on Computational Techniques in Information and Communication Technologies*, pp. 90-96.
Krampen M., Oehler K., Posner R., Sebeok T.A., von Uexküll T. (eds) (1987) - *Classics of Semiotics* - Plenum Press.
Laplante P.A. (2001) - *Dictionary of Computer Science, Engineering and Technology* - CRC Press.
Latino RJ., Latino K.C. (2006) - Root Cause Analysis: Improving Performance for Bottom-Line Results - 3rd edition, CRC Press
Laudon K.C., Laudon J.P. (2009) - *Management Information Systems: Managing the Digital Firm* - Prentice Hall.
Lee J., Lee H., Park J.G. (2014) - Exploring the impact of empowering leadership on knowledge sharing, absorptive capacity and team performance in IT service - *Information Technology & People*, Vol. 27 No. 3, pp. 366-386.
Liu S., Wu B., Meng Q. (2012) - Critical affecting factors of IT project management - *Proc. Intl. Conf. on Information Management, Innovation Management and Industrial Engineering*, vol 1, pp. 494-497.
Maher R.A., Al Hadidi A.M. (2017) - Engineering project management planning and scheduling - *Intl. J. of Civil Engineering and Technology*, Vol. 8, N. 1, pp. 140-148
Martinsons M.G., Chong P.K.C. (1999) - The influence of human factors and specialist involvement on information systems success - *Human Relations*, Vol. 52, No. 1, pp. 123-151.
Firestein S. (2016) - *Failure: Why Science is So Successful* - Oxford Univ. Press.
Monteiro de Carvalho M. (2014) - An investigation of the role of communication in IT projects - *Intl. J. of Operations & Production Management*, Vol. 34, No 1, pp. 36-64.
Nelson R.R. (2008) - IT project management: Infamous failures, classic mistakes, and best practices - *MIS Quarterly Executive*, Vol. 6, No. 2, pp. 67-77.
Okes D. 2019 - *Root Cause Analysis, The Core of Problem Solving and Corrective Action* - 2nd edition, ASQ Quality Press.
Project Management Institute (PMI) (2013) - *A Guide to the Project Management Body of Knowledge* (PMBOK Guide) - Fifth Edition, PMI Available at: https://www.pmi.org/-/media/pmi/documents/public/pdf/about/annual-reports/pmi-annual-report-consolidated-financials-2013.pdf?v=dac03206-adec-43a0-87cf-98ea050138cc
Rocchi P. (2013) - *Logic of Analog and Digital Machines* - 2nd edition, Nova Science Publishers.
Rocchi P. (2016a) - Guest Editorial: Informatics and electronics, some educational remarks - *IEEE Transactions on Education*, Vol. 53, No. 3, pp. 233-239. Available at: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7529250
Rocchi P. (2016b) - What information to measure? How to measure it? - Kybernetes, 45(5), pp.1-16.

Rodriguez J., Esparragoza I., Ocampo J.R., Viganò R. (2017) - Study of motivation of engineering students on multinational design projects - *IEEE Revista Iberoamericana de Tecnologias del Aprendizaje*, Vol. 12, No. 4, pp. 218-226.

Reynolds G. (2018) - *Ethics in Information Technology* - 6th edition, Cengage Publisher.

Sarif S., Ramly S., Yusof R., Fadzillah N.A.A., Sulaiman N.Y. (2018) - Investigation of success and failure factors in IT project management - *Proc. 7th International Conference on Kansei Engineering and Emotion Research*, pp. 671-682.

Schein E.H. (2010) - Organization Culture and Leadership - Wiley & Sons.

Standish Group (2016) - *CHAOS Report* - The Standish Group International Inc. Available at: https://www.standishgroup.com/outline

Steshenko G., Morozov V., Kolomiiets A. (2017) - ”Learning through practice” in IT management projects master program implementation approach - *Proc 9th IEEE Intl. Conf. on Intelligent Data Acquisition and Advanced Computing Systems*, Vol. 2, pp. 935-938.

Streicher T. (1991) - *Semantics of Type Theory: Correctness, Completeness and Independence Results* - Springer

Streit J., Pizka M. (2011) - Why software quality improvement fails (and how to succeed nevertheless) - *Proc. of the 33rd Intl. Conf. on Software Engineering*, pp. 726-735.

Tedre M. (2015) - *The Science of Computing: Shaping a Discipline* - CRC Press.

Thomas G., Fernández W. (2008) - Success in IT projects: A matter of definition? - *Intl. J. of Project Management*, 26, pp. 733-742.

Whitten J.L., Bentley L.D., Dittman K.C. (2003) - *Systems Analysis and Design Methods* - 6th edition McGraw Hill.

Witte D.M. (2017) - Economic evaluation of data warehouse projects - in *Project Management*, Paul C. Tinnirello (ed), CRC Press.