Measuring Financial and Business Operation Impacts, of Software Project Initiatives and Investments, using Project Tracking MarkPoints on the Project Volume.

EVANGELOS MARKOPOULOS
Department of Informatics
University of Piraeus
80 Karaoli & Dimitriou Str., Piraeus
GREECE
epm@unipi.gr

NIKOLITSA BOUZOUKOU
International Business Development Unit
EMPROSS Strategic IT Consultants - USA
One Broadway, 14th floor, Boston, Cambridge, MA 02142
USA
nkb@empross.com

GEORGIOS ALEXOPOULOS
Process Engineering Research Unit
EMPROSS Strategic IT Consultants - Germany
Tumblingerstrasse 54/446, 80337, Bavaria, Munich
GERMANY
gia@empross.com

JAVIER BILBAO
Applied Mathematics Department, Engineering School
University of the Basque Country
Alda. Urkijo s/n., Bilbao
SPAIN
javier.bilbao@ehu.es

Abstract:- Measuring the size of a project has always been a challenge in all the disciplines involved in project management. In software project management, defining a measurement unit for a project is even more difficult since the unique characteristics of the software make it invisible and untouchable, and therefore much more difficult to be measured. This paper presents a model that can contribute towards this issue. The MarkPoint presented can be considered as a sizing and measurement unit to a software project. The MarkPoints are based on the requirements of a project where their initial weighted distribution per requirement, implementation phase and other project elements makes the project size needed for the management of the project. The paper initially states the need for such models in the business world and defines the excepted environment for them to be applied successfully. The model and the overall concept is approach from a business and financial perspective, since it's a business oriented approach driven by business needs and expectations in managing software project and investments.

Key-Words: - Software Economics, Project Management, Project Tracking, Software Sizing, Project Volume Metrics.

1 Introduction

Since the late 60s, even until now, Information Technology (IT) seemed to be the trend of the business world, the competitive advantage, the hidden weapon and the wise infrastructure. Unfortunately those days have gone, and information technology is not a trend to

success anymore but a necessity to survival. Despite the fact that many organizations suffer from lack of technological support or strategy, they remain cautious on technology investments, treating technological projects and initiatives as cost centers instead of profit centers [1]. There are two possible reasons for this behavior. The first one is that organizations do not have the maturity to view the role of technology in the organizational strategy, development and operations with the same respect and confidence offered to the sales or production business units whose financial contribution can be measured quantitatively, but not certainly qualitatively [2]. The second reason is that technology can not, or does not want, to be understood by bureaucratic management and executive boards. People are slaves to whatever they do not understand, and lack of technology drivers can lead an organization to a certain death, sudden or late. Technology and investments in technology are not made for the future anymore but for the present, and if organizations fear such investments then no organizational reengineering, change management, strategy or vision can ever be realized and contribute towards moving the organizations out the crisis which is into consciously or unconsciously[3].

2 Organizational Maturity

The bureaucratic treatment of IT as cost center in an organization have also two other, quite strong supportive arguments.

The first and most important one was, is, and unfortunately will still remain for long, is the bureaucratic management behavior. IT units in many organizations are still under the accounting departments in the best case, or under the procurement department. IT units are still treated as departments, instead of divisions, without budget and representation in the board of directors. Lack of strong IT representation on the Board of Directors with significant role in organizational strategy, doesn't allow the role of IT and the need for IT investments to be understood and justified by the management executes at high level[4].

A second argument is that most of the organizations do not have the required organizational maturity in personnel and procedures that will clearly understand and support IT initiatives and investments [5]. It is quite common in nowadays to have organizations willing to move to technology but feel unable to do so, since the human recourses and organizational infrastructure cannot be aligned with the maturity required to adopt technology and successfully integrate it in the business operation and production environment.

Organizational maturity is a concept of continuous business and operations research. Organizations suffer not because they cannot solve their problems but because they cannot understand them [6].

Organizational maturity is the alignment of the people with the organizational goals, vision, and objectives using processes, standards and best practices as knowledge injections to the organizational units that need it the most. Inspiring people is the beginning towards organizational maturity and infrastructure, setting up to the visions and strategies that will be called to be applied [7].

3 The Concept of Technocracy

Organizations are being developed today either technologically or technocraticaly.

The technological development of an organization is based on the transformation of the organization using state of the art technology on all or almost all of its activities. On the other hand the technocratic development of an organization is based on the use of the technology needed at a given time, and being capable to be adopted and used successfully. In other words a technocrat is not being carried away by fascinate technological break-troughs and state of the art ideas, products and concepts. A technocrat has a more conservative profile taking one step at a time, or making break a through, knowing that he is capable to carry them out successfully. Today all organizations have to be technologically transformed, meaning that they have to use technology into their everyday activities or else they will die. Today organizations need to totally redirect their workflow process to integrate technological support in all of their operations [8]. Organizations need not to move slowly on technology if they have the capability to go faster. Unfortunately not all organizations can adopt technology successful at the same pace, and achieve same high rates on return of their investments. For some organizations, that might be a challenge, since they have the resources to deal with a possible productivity curve, but for others that could be impossible, or sound impossible, at least for a specific period of time.

Technocratic organizations are being built by investing on the development of the proper technocratic maturity within the organization that will be capable of utilizing and using the technological capabilities of every technology being adopted. A technologically mature organization is the one that knows its capabilities and aligns them with its goals, or aligns the goals if needed.

Organizations at low technocratic maturity levels use basic or simple project management processes to secure its technological investments and efforts, while

organizations in the high technocratic maturity levels used advanced project management activities that can assure and predict the success of everything they decide to attach on technology [9].

4 Measuring Technocracy

The degree of technocracy in an organization promotes organizational development initiatives which are actually the proper insertion and adaptation of the technology to the organizational production and operations. On the other hand technocratic development needs to be performed by measuring the effectiveness of the organizational goals, human recourses and technology. These three elements, whose relationship is presented at figure 2, are the most critical towards organizational technocratic development.

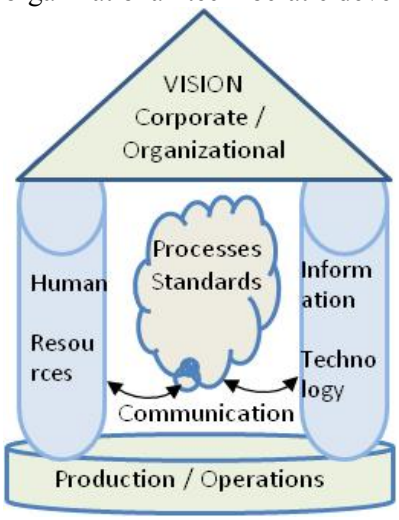

Fig 2. Technocratic Maturity Concept.

The most important element in technocratic development measurements is the existence of realistic goals and visions. The term realistic might sound quite fuzzy, but it is not, if it can be aligned with the other two elements which are the people and the technology. If an organization is based on people with no vision and/or basic technocratic mentality then the organization is low in technocratic measurement. Low technocracy is also in the organizations with people capable, willing to adopt any technology that will improve the organizational production and operations, but have insufficient technological infrastructure.

Vision without people and technology is wishful thinking. Likewise people without technocratic mentality and a vision to lead them are the same as people with technocratic mentally but without technology to use and a vision to lead them.

In order to manage the balance on the three key technocratic development elements, the process comes to take the role for bridging them. By measuring the technocratic processes, an organization can be measured against specific technocratic models and metrics. Such technocratic management models can be considered the process models CMM (Capability Maturity Model) [10] and the current CMMI (CMM-Integrated)[11], the P-CMM (People-CMM)[12] and also a number of other models and concepts dedicated to the improvement and measurement of technocratic development initiatives. Such concepts are the Personal Software Process [13], the Team Software Process [14], Mutational Process Models [15], the BITS (Balanced IT Scorecard) [16], and others that incorporate the balance of vision, technology and people in process management.

5 Managing Technocratic Initiatives and Investments

Being technocratic, requires being organizationally mature. Taking the definitions provided in the previous sections backwards, it can be indentified that the prime reason organizations are not getting mature and therefore technocratic is the fear they have for the technology. This fear is not actually technical but more operational, especially when it comes to software projects, initiatives and investments which sound complex to the business.

Software is a brain product, it cannot be touched, measure and evaluated quantitatively during its acquisition and development process. Therefore all software projects are somehow judged after they have been delivered. Software project failures tend to be common stories in the business world regardless the size of the organization. Software projects and investments failure can happen to anyone with no technocratic mentality and that is quite scary.

The United States General Accounting Office after inspecting the US Department of Defense, found that from the many software projects which worth nearly 9 billion US dollars at 1982, only 5% of them were actually delivered and operate, even with changes in the delivered code. The rest of the projects were either paid but never delivered, operated for a short period of time and then were abandoned, or delivered but never operated (fig 3).

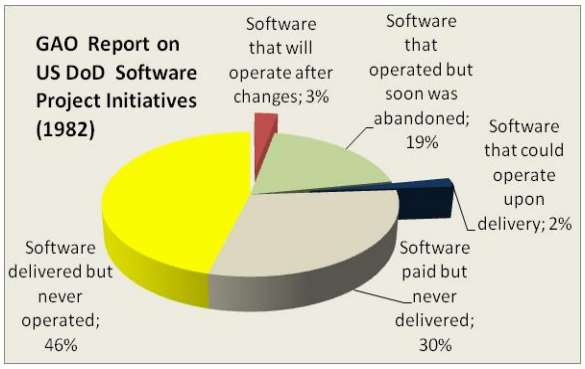

Fig 3. GAO-USDoD Software Projects Evaluation

Likewise, in 2000, GAO reported for NASA that the space agency has continued to use "undefinitized contract actions," under which NASA authorizes work to begin before the final estimated cost and fee is established through negotiation. The report found that the NASA methodology of "faster-better-cheaper" has failed to sustain a good management environment, and has resulted in inadequate insight and oversight. Poor information transfer and inadequate cost margins have left the agency vulnerable to unexpected cost overruns and design failures. The major cause for these situations was that the contracts NASA developed were not well defined in both the subject of the project, and the implementation estimations. NASA officials could authorize work to begin on a contract change before NASA and the contractor agree on a final estimated cost and fee. Such changes are referred to as undefinitized contract actions - that is, unnegotiated contract changes. GAO found NASA had made 593 changes totaling $8987.7 million during fiscal 1998 and 1999 in its prime station contract with Boeing[17].

The US-DoD and NASA justify the software crisis which began in the early 80s and seems to exist in nowadays [18]. Maybe not as intensive as when it started but it is still present making all type and size of organizations cautions on their technological investments [19]. For this reason the software project management international community invested billions of dollars and staff power towards the development of process models, methods, standards, guidelines and best practices that could manage, if not stop this software crisis.
Organizations fear being technocratic since they fear the outcomes of an IT project failure [20].
Managing software projects is a totally different ball game for most of the organizations worldwide. Many take the risk to remain bureaucratic as long as they can survive while others, more brave and with solid financial background, or not, take the initiative to invest on IT knowing that the management of the IT projects can possibly cost more than the project itself if is development won't be performed properly. Budget overruns, time overruns, unstable requirements and many other horror issues usually multiply the original budget of the project that was not being able to be determined in the first place.
Since all those issues and problems are considered part of the software crisis, solutions do exist, and there is no need for an organization to be rich in order to be technocrat, and survive in today's business world driven mostly by bureaucrats.

6 Project Management via Requirements Management

A way out of the software crisis is the management of the requirements that form the project [21]. The requirements of the project are the definition of the project. More precise the requirements have smaller the surprises in the development process and also later in the operations of the software as well.
Most of the software projects fail on the requirements phase which is the first phase on the implementation route of a project. The requirements in a project can be considered as living organisms in within project. The requirements formulate the project, and their implementation progress guides the implementation process of the entire project [22].
Project management by requirements has always been considered a revolutionary project management approach, if and only, the requirements can be clearly defined and remain stable for time intervals that will not affect the development of the project. Since this cannot be absolutely possible, requirements changes are grouped and are implemented as enhancements in system versions either in defined time phases or based on the volume or criticality of the requirements lined up to be implemented.
Managing the requirements can be considered as managing the project or the investment lifecycle. The requirements will formulate the project goals and objectives, will identify the project deliverables and will be the base for building a solid tender. The responses to the tender from the candidate suppliers will be based on the tender requirements, which are the project expectations. Complying to requirements means replying to the project. Based on the tender replies, the tender winner can sign a contract according to the proposed tender reply which is nothing more than compliance to the project requirements. The requirements based contract will be the key tool to perform project management by contract, if needed. Suppliers knowing that their contracts are requirements based, perform requirement based project implementation [23].

Taking the requirements further down the project implementation lifecycle, they end up affecting the project life cycle since they can also have an impact on the maintainability strategy, policy and contracts. Figure 4 presents the role of the requirements on the management of software projects, initiatives and investments.

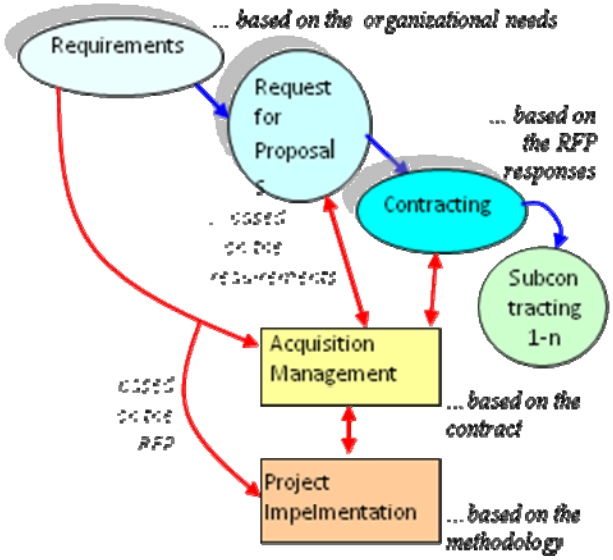

Figure 4. Requirements based project management.

7 Sizing Software Projects in Cost

In order to manage a project successfully it must be sized properly. Project sizing, towards project management has been an issue of advanced project management and software engineering research.

Besides the COCOMO which still remains a model that can predict quite successfully the size of a project in terms of development effort, most of the other software sizing models can be considered as either too engineering or too technically specific [24]. Even the COCOMO, in order to define the project implementation effort it incorporates other engineering practices that defined the project complexity on which the effort is being calculated after that. The COCOMO incorporates in its calculation models many techniques, but most of all the function point analysis that determines the complexity of the project functionality.

Based on the complexity of the project functions and value multipliers on each function, the model generates the effort and time to develop it. The effort can them be translated into staff months based on the expertise required to implement it and the cost based on staff-month rate per expertise (eq1).

$$\text{Staff-Month (SM)}: \quad SM = a * (KDSI)^{b} \quad \text{(Delivered Source Instructions)} \qquad \text{(eq.1)}$$

$$\text{Time of Develop (TDEV)}: \quad TDEV = c * (SM)^{d}$$

Project tracking and management based on the COCOMO can be performed by tracking the project implementation costs in order to reach the project cost identified initially by the model which was calculated by the complexity intensity of the project functions and not by its requirements. Therefore project management by requirements cannot be effectively performed by the COCOMO which treats the project more from a technical perspective than a business one.

8 Sizing a Software Project in Project Weight (MarkPoints)

Treating a software project businesswise can be quite difficult, since software projects are considered complex by nature and engineering complexity is oppose the business logic and expected simplicity towards decision making.

In order to move the complexity from the engineering dimension to the business dimension, software projects and initiatives need to be measured against metrics that have no engineering flavor and can be understood practically, otherwise they will be called ‘theoretical’, like they always do when they cannot be understood by executives.

A project can be managed much more easily if it can get a volume weight, where the implementation process of the project will be measured by either reaching the project weight or eliminating to 0.

In order to create this project weight the project requirements can be used as a metric base.

This volume can be measured by MarkPoints (points that Mark the progress of the project or MARKopoulos Points).

The total weight given by the project requirements makes the project volume in MarkPoints. The requirements are categorized in Single Requiremnts (SR) and Requirements Groups (RG). The Requirement Groups are used in order to achieve accuracy on sizing the project by MarkPoints. Each SR or RG can have weight (SRW and RGW) based on the criticality of the requirement that it carries on the total project. A requirement group contains one or more single requirements (eq2)

$$Prj = \{ SR_1, SR_2, SR_3, \ldots SR_n \}$$

$$\text{(eq.2)} \qquad Prj = \{ RG_1, RG_2, RG_3, \ldots RG_n \}$$

$$\text{where : } RG_1 \neq RG_2 \neq RG_3 \neq \ldots \neq RG_n$$

Each requirements group has a weight to the total effort and criticality of the group to the total project. The sum of the requirements groups weights (RGW) is equal to the total weight of the project (eq3.)

$$\text{Project completion} = \text{(eq.3)} \quad = \frac{\sum_{i=1}^{n} \frac{(SR_i W_j)}{n} * 100}{\sum_{i=1}^{n} (SR_i W_j)}$$

Taking the precision of the measurement down to even more accurate and also realistic levels, the MarkPoints of a project are calculated by integrating the number of project phases and their weights.

That means that if a project has an implementation methodology with ten development phases then the number of the implementation phases and their complexity will be added to the project total weight.

Therefore the total MarkPoints of a project is the sum of the project requirements weights multiplied by the project implementation / development phases (DP), giving the project total points (eq. 4).

$$\text{(eq.4)} \quad \sum_{i=1}^{n} (SR_i W_j) * \left(\sum_{i=1}^{n} (SR_i W_j) * \frac{(DP_i)}{10} \right)$$

This approach supports the project tracking progress more accurate since not all implementation phases have the same complexity and therefore the completion of some implementation phases with low complexity might not equal the completion of an implementation phase with high complexity.

9 Applying the MarkPoints Project Measurement Unit

Having defined the size of the project, allows project management to be based on project progress readings through project tracking inspections.

The MarkPoints can be applied only when a structured project tracking models are applied. Such tracking models need to be requirements oriented since the MarkPoint is a requirements based measurement unit. The computations of this paper are based on the MBA-SPI (Metrics binder Analysis for Software Projects Initiatives) Tracking Model [25]. Having defined the MarkPoint measurement unit, it can be applied on structured progress project metrics with similar characteristics. The MBA-SPI model can be considered as precondition of the MarkPoint measurements since they both share the same measurements structure, elements and measurement readings. The MarkPoints on the other hand can be adjusted to any other project tracking model, not necessarily the MBA-SPI, that needs to be in place and used by either the project management or the customer/client organization.

The MarkPoints do not represent a tracking process but a project measurement unit that can be used by any project tracking model which performs structured and requirements project tracking based on measurements reading at a certain frequency dictated by the project goals.

The project progress readings are performed on time intervals defined by the project implementation period or by the project manager based on the criticality of the project or the accuracy desired.

Using the MarkPoints a project manager has the capability at any instance of the project implementation period to calculate the project progress by identifying the MarkPoints gained so far in the implementation process (eq5).

$$\text{(eq.5)} \quad 100 - \left[\sum_{i=1}^{n} (RG_i) * \left[\sum_{i=1}^{n} \left[\left(\sum_{i=1}^{n} (DP_i W_j) \right) + \right. \right. \right.$$

$$\left. \left. \left. + \left(\frac{\sum_{i=1}^{n} (SR_i W_j) * \left(\frac{Comp_i}{10} \right)}{\sum_{i=1}^{n} (SR_i W_j)} * \left(\frac{RG_i W_j}{10} \right) \right] - RG_i W_j \right] * (-1) \right]$$

The remaining implementation MarkPoints create the difference to project completion and can be interpreted as remaining distance to target MarkPoints or estimate a completion percentage.

Besides the overall project progress status, the MarkPoint model can provide progress information per requirement at a given project implementation phase (eq6).

(eq.6)

$$\sum_{i=1}^{n}(RG_i) * \left[\sum_{i=1}^{n} \left[\left(\sum_{i=1}^{n}(DP_iW_j) \right) + \frac{\sum_{i=1}^{n}(SR_iW_j) * \left(\frac{Comp_i}{10} \right)}{\sum_{i=1}^{n}(SR_iW_j)} \right] * \left(\frac{RG_iW_j}{10} \right) - SR_iW_j \right]$$

or per requirement group at a given implementation phase (eq7).

That means that the completion of the requirements in a requirement group i, in phase j, consumed x MarkPoints with y the maximum MarkPoints to be consumed for the specific phase. The difference between Y and X indicate the absolute implementation success of the requirements group, else it indicates that a one or more requirements did not reach 100% completion in that phase.
Being able to identify such requirements completion behavior per implementation phases it is easy and valuable to track the overall completion progress of each requirement, requirement group or project (by adding all requirements) at any instance.

In order to be more precise the term 'instance' does not need to be restricted to the project implementation phases. If the measurements were done per project implementation phases then the project tracking readings were to be performed once per implementation phase. These wide time intervals between inspections and tracking readings do not give the model and the project management effort the accuracy and confidence required to take the management decisions needed in order to place the project back in track, if deviations from the plans are identified.
The tracking inspections need to be executed per short time intervals in a project implementation phase. Weekly inspections are recommended. Biweekly inspections can provide more accuracy but take much effort to be performed.
In weekly metrics, the project manager can identify the MarkPoints consumed by the project implementation process that specific week, and by identifying the remaining MarkPoints needed to be spend towards completing successfully the current implementation phase. Thus the overall project management turns to be a MarkPoint spending issue, translating project progress success into monopoly money that need to be spend at the end of the project implementation process.

The MarkPoints have been designed in such way that can give volume to the project size. They aim to be established as a project measurement unit. Projects can be measured in MarkPoints and managed by tracking the project implementation progress, which means spending them trough out the project implementation period.

(eq.7)

$$\sum_{i=1}^{n}(RG_i) * \left[\sum_{i=1}^{n} \left[\left(\sum_{i=1}^{n}(DP_iW_j) \right) + \left(\frac{\sum_{i=1}^{n}(SR_iW_j) * \left(\frac{Comp_i}{10} \right)}{\sum_{i=1}^{n}(SR_iW_j)} \right) * \left(\frac{RG_iW_j}{10} \right) - RG_iW_j \right] \right]$$

MarkPoints make the concept of project management and project progress measurement quite simple and fun in a way.
On the other hand MarkPoints can be considered quite reliable since they take into consideration all the following factors: i) project requirements, ii) requirements complexity, iii) requirement criticality, iv) requirement group criticality, v) project implementation phases, vi) weight of project implementation phases.
The combination of all the above factors create unlimited interpretations of the MarkPoints measurements and unlimited metrics as well.

10 Interpretation of the MarkPoints in Project Management Goals

Integrating simple and complex statistical analysis methods in the MarkPoint measurements can provide quite impressive results by analyzing the distribution of the MarkPoints per requirement, requirement group, measurement period, implementation phases and time overall.
Such measurements and observations can identify project risks in the implementation process, or risk trends that can be found later in the project operations period.
Likewise similar findings can be identified in the quality of the systems being developed, the reliability of the system, and other areas of project management, investment and quality assurance that can turn the MarkPoints into a general investment management model.

Since software projects are generally characterized as technological investments, due to their cost and impact in an organization, the MarkPoints also can be used towards calculating other project management and investment management is critical area areas as well.
The MarkPoints can be used in project cost management or project costs in general.
Project cost through MarkPoints can be obtained by assigning the implementation of each requirement to a systems developer. According to the complexity and weight of each requirement, the cost of the requirement can be associated with the cost for implementing the requirements which can be associated within the expertise of the system developed, and the time required to implement it.
Thus the MarkPoints can be translated to implementation effort based on the requirements critically, which can be translated to staff power which can be translated to implementation cost (fig 5).

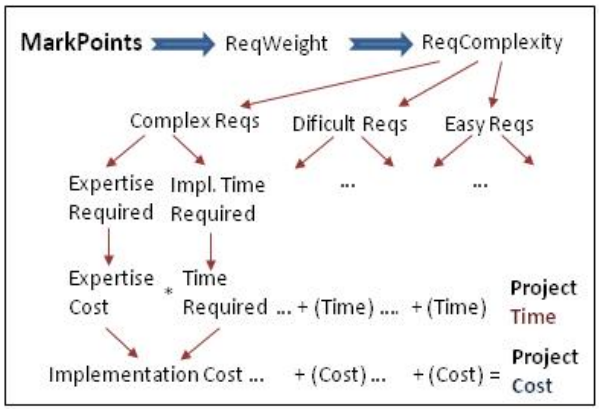

Figure 5. Project Cost Estimation Using MarkPoints

Besides estimating the cost of a project using the MarkPoints, estimations can also be performed on risk management and quality assurance areas. If for example, there are MarkPoints left over on a specific progress reading, and add up to more and more MarkPoints over the readings after, the specific requirements or requirement group need to be investigated in terms of engineering quality or implementation complexity that could generate risks from those delays sooner or later. The interpretation of the MarkPoints have no limits as unlimited are the project goals and expectations. If the goals of a project are financial ones, the MarkPoints can be used towards measuring the cost distribution or the cost estimation. If the project goals are short time to production and operation, then the MarkPoints can be used towards measuring the elapse time to completion, and so on.

11 Pre and Post Conditions on using the MarkPoints

The pre and post conditions to apply the MarkPoints successfully can be characterized in three categories.
The first category is the management of the project requirements. The successful operation of the MarkPoints on a project is not the effort required to set them up and to distribute them on the project requirements, but the identification of the requirements themselves.
The most critical preconditions on using successfully and reliably the MarkPoints is the development of a clear and well defined set of project requirements. The requirements does not need to be functional but non functional as well in order for the MarkPoints to provide the maximum accuracy. Non functional requirements take implementation time and effort, not measured by transaction of function completion. Identifying the non functional requirements is another pre condition of the MarkPoints.
Besides the requirement identification effort, the rest of the MarkPoint pre conditions are related with the personnel that will use the MarkPoint model.
The second category of pre and post conditions using the MarkPoints is the project management team that will support this initiative.
Giving value to the requirements and identifying their criticality to the project goals, is task for a domain expert. If a project is in the banking sector an experienced banker needs to be involved in this requirements weighting task. Also experienced software engineers required to determine the weight of the implementation phases, and the number of them that will be used towards the implementation of the project. Along with the software engineers, experienced project managers are expected from all the parties involved, in order to agree on the inspection and measurements results at one. No time need to be wasted on arguments for completion rates of requirements at a given time and phase. Projects with hundreds of requirements need to be inspected as fast as possible.
The third category of pre and post conditions
towards using the MarkPoints is the organizational maturity.
The management of all the organizations involved in the project need to participate in the MarkPoits results and analysis which is generated right after each measurement period. This management involvement is expected primarily from the project developer (supplier) and the project owner (customer). The MarkPoints generate results, trends and analysis that affect mostly the customer and then the supplier of the project. If there is not technocratic mentality on the personnel from the customer's side primarily, then the results of the

metrics will not trigger any actions, decision or initiatives towards correcting any deficiencies or deviations in the expected quality or implementation schedule.
Organizational maturity and technocratic maturity are the most critical preconditions and post conditions as well, expected primarily from the customers' part. After all, the project will be accepted and used by the customer, and therefore the responsibility to judge the progress of the project and take the right decisions belongs to the customer.
Figure 6 presents the categories of pre and post conditions towards using the MarkPoints successfully.

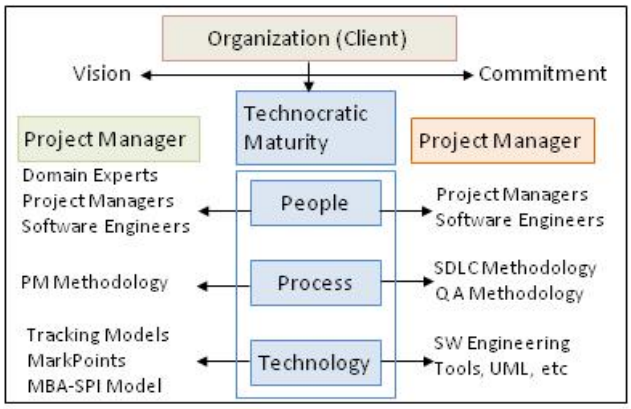

Figure 6. MarkPoints Pre and Post Conditions

12 Areas for Further Research

The MarkPopints is a new measurement system. Like the kilos used to measure mass and the gallons to measure liquids, the MarkPoints are used to measure the size of a project.
A project is of size X MarkPoints. This project sizing can be used as a project management effort identifier to other projects with similar MarkPoints values. Thus a project baseline can be generated where project post completion analysis reports can be store and categorized based on their MarkPoints value. When a project is retrieved all the implementation effort, process, progress, risks and other information will be available and can be used to manage new projects with similar MarkPoints values [26].
This software engineering and project management repository is an area of further research. Even that the MarkPoints have already being practically applied in large scale projects, there are not enough projects in the project baseline to be used for such estimations and management guidance. Until then, the research conducted towards reaching this goal will continue.
Another area of further research is the actual interpretation and transformation of the MarkPoints to project cost estimation and cost management modeling, as well as the modeling of the MarkPoints to be applied as successful to the project risk management, defect prevention and quality assurance disciplines [27].
The MarkPoints model conceived and developed by EMPROSS Strategic IT Consultants (www.empross.com), an international organization specialized in Organizational Technocratic Development Strategies and Technocratic Investment and Initiatives Management Models.
The development of the model was a result of the continuous evolution of the ARIADNE Methodology for Technocratic Project Management, developed by EMPROSS as well. The need for this evolution of ARIADNE derived after studying the results of many project management projects managed by EMPROSS. Further research on the MarkPoints will be supported not only by academic challenges but primarily from business challenges.

13 Conclusion

In order for the MarkPoints to be executed successfully, and the involved organizations to benefit from the usage of such models, the organizations need to be technocratic. A vision must exist, along with the people willing to participate in vision reaching initiatives supported by technology.
Technocratic maturity requires organizational maturity meaning that in order for an organization to be technocratic it needs to have and trust the technocrats in the board of directors. It is important for technocrats to have sound voice in corporate decision making sessions, as well as the existence of autonomous IT business units that must not be treated as cost centers to the organization but rather as profit centers or lifesavers.
The MarkPoints is a revolutionary approach on software project initiatives and investments management. They are solely based on the requirements management and the requirements process.
The requirements of the MarkPoint Model require open mind and wide thinking in order to perform all the actions that will set up the environment for the MarkPoints to be applied effectively. After all nothing is free today. There is not a fee ride anywhere, but some rides can be considered quite cheap if the proper preparation is in place.
The MarkPoints is a new project management concept incorporating not only engineering methods and techniques but rather an organizational and business philosophy quite important in our days with the financial crisis all over the business world.

References:

[1] Martin J. 'The Great Transition', Amacon Press, 1996

[2] Olve N-G,. Roy J., and Wetter M., 'Performance Drivers: A practical guide to using the Balanced Scorecard', John Wiley, 1999

[3] Deming W. E., 'Out of Crisis', MIT Press, 2000

[4] Bergey J., Smith D., Tilley S., Weiderman N., Woods S. Why Reengineering Projects Fail. Technical Report, CMU/SEI-99-TR-010, ESC-TR-99-010, April 1999

[5] Markopoulos E., Bilbao J., 'Process Development Management towards Organizational Maturity', International Journal of Computers, Vol 4, Issue 22, 2008,pp361-367

[6] Gartner, J. 'Renewal of Organizations.', 20th Annual Meeting of the Board of Trustees, Midwest Research Institute, Kansas City, MO. May 3, 1965

[7] DeMarko T., Lister T., 'Peopleware, Productive Projects and Teams', 2nd Edition, Dorset House Publ. 1999.

[8] European Commission. 'The business benefits of software best practices', Case Studies, ESSI office, EC, N105-3/43. Brusseles, 1996

[9] Humphrey W., 'Managing the Software Process', Addison Wesley, 1995

[10] Paulk Mark C., Curtis B., Chrissis M. B., Weber C. V., 'Capability Maturity Model for Software, Version 1.1', Software Engineering Institute, Carnegie Mellon University Pittsburgh, PA, Technical Report, CMU/SEI-93-TR-024, ESC-TR-93-177, February 1993

[11] Chrissis M. B., Konrad M. Shrum S., 'CMMI Guidelines for Process Integration and Product Improvement', Addison Wesley, 2003.

[12] Curtis, B. et al, 'People Management Capability Maturity Model', Software Engineering Institute, Pittsburgh, PA, 1994.

[13] Humphrey W., Introduction to the Personal Software Process', Addison Wesley, 1997

[14] Humphrey W., 'Managing Technical 4People: Innovation, Teamwork, and the Software Process', Addison Wesley 1996.

[15] Markopoulos E., J. Bilbao, T. Stoilov, T. Vos, C.F. Talamanca, 'Project Management Stage Mutations within Agile Methodological Framework Process Transformations', WSEAS Transactions on Information Science and Applications, Volume 5 , Issue 5, May 2008

[16] Reo D., Quintano N. & Ibanez M., ESI Balanced IT Scorecard Process Perspective V1.1, European Software Institute, ESI-1999-TR-016, February 1999

[17] NASA – General Accounting Office Report 'Major Management Challenges and Program Risks: National Aeronautics and Space Administration' (GAO-01-258), 2000

[18] Gibbs W., 'Software's Chronic Crisis', Scientific American, vol. 271, no. 3, pp 86-95, 1994

[19] Glass, R., 'Is there really a software crisis', IEEE Software, vol. 15, no. 1, January 1998, pp.104-105

[20] The CHAOS Report, The Standish Group International, Inc., 1995.

[21] Anton I., Successful Software Projects Need Requirements Planning, IEEE Software, Volume 20, May-June 2003, pp 44-46.

[22] Jones Capers, 'Patterns of Software Systems Failure and Success' 1996

[23] Robertson S., Robertson J., Mastering the Requirements Process, ACM Press, Addison Wesley, 1999

[24] Boehm B., 'Software Cost Estimation in COCOMO II', Prentice Hall, 2000

[25] Markopoulos E., G. Alexopoulos, N. Bouzoukou, J. Bilbao., 'Project Tracking using a Metrics Binder Analysis (MBA) model on Software Project Initiatives (SPI)', Proceedings of the ACC09-Applied Computing Conference- WSEAS, 2009

[26] Humphrey W., 'A Discipline for Software Engineering', Addison Wesley, 1995

[27] Keil M., 'A Framework for Identifying Software Project Risks', CACM vol.41, no.11, November 1998, pp 76-83.