Software Project Volume Sizing for Tracking Software Intensive Investments and Business Initiatives

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: - Software Project Management has always been considered as a Herculean task for organizations and enterprises with limited or insufficient technical expertise and resources. The difference contradiction between business and engineering conceptions on project definition, management and success, resulted in project failures and catastrophic technological investments and business initiatives. This paper aims to contribute in software project management by identifying a new innovative measurement unit based on which software projects can obtain a volume that will be used throughout a project tracking process integrating business and engineering goals and objectives under a common project success perception. This new project tracking approach can assure financial and operational benefits by minimizing the lack of communication between business and engineering parties involved in the implementation of a software oriented investment or business initiative.

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

1 Introduction
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 fascination technological breakthroughs and state of the art ideas, products and concepts. A technocrat has a more conservative profile taking one step at a time, or making breakthrough, 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. Organizations need to totally redirect their workflow process to integrate technological support in all of their operations [1], moving slowly on technology only if they do not have the capability and the maturity to go faster. Unfortunately not all organizations can adopt technology successfully 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 [2].

Technocratic development needs to be performed by measuring the effectiveness of the organizational goals, human resources and technology [3]. These three elements, whose relationship is presented at figure 1, are the most critical towards organizational technocratic development.

![Technocratic Maturity Concept](image)

Fig 1. Technocratic Maturity Concept.

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 [4].

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 [5].

2 Technocratic Business Initiatives and Investments Management

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 [6].

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 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.

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. Such changes are referred to as undeﬁnitized contract actions - that is, no negotiated 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 [7].

The US-DoD and NASA justify the software crisis which began in the early 80s and seems to exist in nowadays [8]. Maybe not as intensive as when it started but it is still present making all type and size of organizations cautious on their technological investments [9]. Organizations fear being technocratic since they fear the outcomes of an IT project failure [10]. 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.

3. 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 [11]. 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).

\[
Staff-Month (SM) : \quad SM = a \cdot \{KDSI\}^{b}
\]

(eq1)

\[
(\text{Delivered Source Instructions})
\]

Time of Develop (TDEV); \[TDEV = c \cdot (SM)^{d}\]

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.

4 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 [12].

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 Requirements (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)

\[
Prj = \{SR_1, SR_2, SR_3, \ldots, SR_n\}
\]

(eq.2) \[
Prj = \{RG_1, RG_2, RG_3, \ldots, RG_n\}
\]

where: \[RG_1 \neq RG_2 \neq RG_3 = \ldots = RG_n\]

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.)

Project completion =

\[
\frac{\sum_{i=1}^{n} (SR_i W_i)}{\sum_{i=1}^{n} SR_i W_i} \times 100
\]

(eq.3)

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).

{eq.4} \[ \sum_{i=1}^{n} (SR_iW_i) + \left( \sum_{i=1}^{n} (SR_iW_i) \times \frac{(DF_i)}{10} \right) \]

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.

5 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 [13]. 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).

{eq.5} \[ 100 \times \left( \sum_{i=1}^{n} (SR_i) \times \frac{(DF_i)}{10} \right) + \left( \sum_{i=1}^{n} (SR_iW_i) \right) + \left( \sum_{i=1}^{n} (SR_iW_i) \times \frac{(Comp_i)}{10} \right) \frac{(SR_iW_i) - (SR_iW_i)}{-1} \]

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.

6 Using the MarkPoints towards Measuring Project Completion

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

Toward being 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.

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.

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.

7 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 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 2).

![Figure 2. Project Cost Estimation Using MarkPoints](image)

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.

8 MarkPoints Usage Requirements

Using the MarkPoints successfully a number of conditions needs to be in place as usage requirements based on them the model will perform more effectively. Theses model usage requirements can be characterized in three categories.

The first category is the management of the project requirements (functional and non). 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 [14].

The most critical MarkPoints usage requirement is the development of a clear and well defined set of project requirements. Besides the project requirement identification effort, the rest of the MarkPoint usage...
requirements are related with the personnel that will use the MarkPoint model.
The second category of the MarkPoints usage requirements is the project management team that will support such an initiative. 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.
The third category of the MarkPoints usage requirements is based on the organizational maturity. The management of all the organizations involved in the project need to participate in the MarkPoints 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 [15].

9 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: