Project Tracking using a Metrics Binder Analysis (MBA) Model on Software Project Initiatives (SPI)

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

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

NIKOLITSA BOUZOUKOU
International Business Development Unit
EMPROSS Strategic IT Consultants - USA
One Broadway, 14th floor, Boston, Cambridge, MA 02142
USA
nkb@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:- Project management and software project management in particular is about managing the implementation effort required towards the completion of a project in time, budget and quality. The management effort is based heavily on project tracking techniques and practices were the project implementation progress is closely monitored and analyzed. This paper presents a project tracking model based primarily on the project requirements and its implementation evolution through the project lifecycle. The tracking process is supported by a set of metrics which cumulatively collaborate, since day one of the project implementation, towards the interpretation of the real project progress via various measurements and results continuously. This live project tracking system is a metric binder analysis for software projects and initiatives, were more than 30 metrics are bind together, creating an accurate, practical and realistic picture of the project progress for the entire project or for any of its components, all the way down to its requirements.

Key-Words: - Software Metrics, Project Management, Project Tracking, Software Quality Engineering

1 Introduction

Managing a project and in particular software oriented projects has always been a challenge. [1],[2]. Software project management is primarily based on understanding the nature, scope and goals of the project in order to perform the proper allocation of resources needed towards its successful implementation management [3], [4].

A very important but trivial management principle is the definition of success. Success can be defined by the deviation of the project results and outcome from the expected ones [5]. The expectations on the other hand can be defined through a variety of approaches that can be considered technical, financial, operational and organizational [6]. A project with budget overruns might be more successful than a project within budget but with functional deficiencies [7]. The balance between the project success elements has always been considered as an interpretation of the definitions of success.

Despite any type of success definition, in order to reach a result, successfully or not, a metrics system needs to be in place based on which all the success elements will be tracked and evaluated [8]. Project management has solely been based on project tracking initiatives and approaches that can generate the required project picture needed for decisions to be taken. Unfortunately there are contradictions in this subject too. What needs to be measured, how it will be measured, and what is the value of the measurements, remains still an open issue in the international project management community.

Many project tracking models have been generated, in software tracking as well. Some of them are too theoretical or complex to be applied in small to medium size projects, and others have been too generic, requiring special interpretation in order to set up the metric values and expectations per project. Trying to measure a software project quantitatively is as complex trying to measure it qualitatively. Subjective measurements are part of every metric system but the level of subjectiveness is mat makes each tracking model and valid system, valid and successful and acceptable [9].

2 Software Project Risks

Software projects are quite different from any other type of project. Software is considered a brain product, with no real value and weight, but with only virtual, characteristics that can be quantitatively measured and tracked [10]. Software development processes and methods are overwhelmed with this fuzziness which is being passed on the software project tracking models as well. If a project is not well defined then it is impossible to be tracked and measured [11].

Project metrics existing in other project management areas of application (manufacturing, civil engineering, etc) are absolutely absent from the software project management application. Software project management hides many critical risks not only to the project, but also to the organizations involved in the development. That includes the customer as well. A project failure is everyone's failure. It's a failure of the project manager, unable to control the project, of the project developer unable to manage the requirements, and also a failure of the project owner unable to provide requirements [12], or make decisions. If what is being developed is unknown, it is impossible to be tracked and therefore to be managed. All project risks, not only in the project implementation periods but in its operation as well, start and end at the requirements. The requirement is the critical success factor in software project management and software project tracking as well [13], [14].

3 Measuring the Progress of a Software Project

Metrics and measurements sound similar but are actually absolutely different concepts. Metrics defines what needs to be measured and measurement defines the metric applying process. Like every metric model, accuracy derives from the quantity and quality of measurements.
In order to practically understand the qualitative and quantitative aspects of metrics and measurements, a Metrics Binder Analysis (MBA) model has been developed and to be applied in particular on Software Project Initiatives (SPI) as the prime application target group.

The MBA – Software Project Initiatives (SPI) is a software project tracking model heavily based on the requirements of the project. The model actually transforms the project requirements into requirements groups and manages the requirements implementation based on the implementation of each requirement, requirement group in respect to time which is calculated based on the implementation methodology phases and time to execute each implementation phase. Figure 1 presents a high level approach of the MBA-SPI model elements.

Unlike other project tracking models, the MBA-SPI can be considered quite subjective and adjustable to any definition of project success, due to the weighting system that is being applied to all the model elements. Weights are given to the requirements groups, the requirements themselves, the implementation phases and to the overall project. The weights can vary from element to element, meaning complexity, criticality, or progress. This mutational project success definition approach is what makes the MBA-SPI model adjustable to all type of software projects regardless the criticality, importance, complexity, size, strategy or technology behind them.

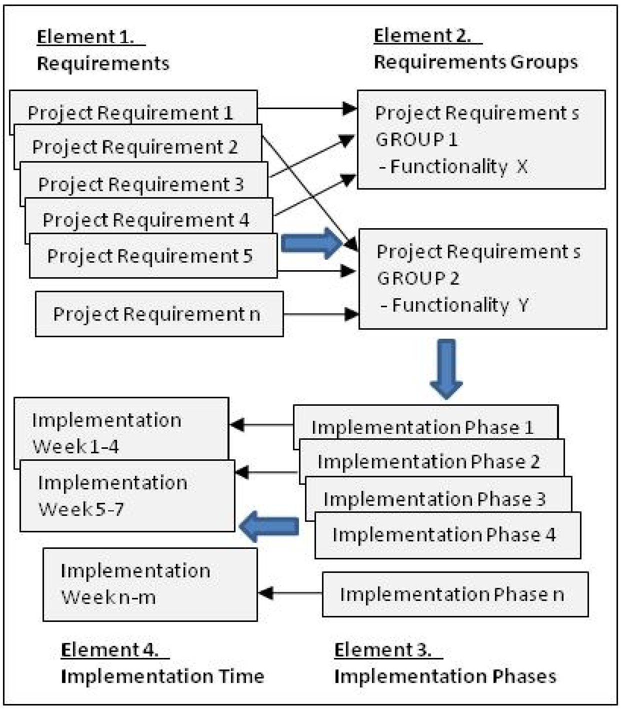

Figure 1. MBA-SPI Model Elements.

4 The Metrics Binder Analysis Model for Software Project Initiatives (MBA-SPI).

The MBA-SPI is based solely on the project requirements. A well defined set of requirements is the base of the MBA-SPI model. If an organization or project manager can define the project requirements then no method, standards, best practices or technique in software engineering, software quality, and project management that can ever be applied. Requirements management is the first step which can be made from the chaotic project management to the structured, agile and mutational project management [15] if we want to go that far. Lack of requirements management mentality is equivalent to lack of project management capability, understanding or principles.

The MBA-SPI model initially categorizes the project requirements into project requirements groups (RG). Each group defines system functionality, quality assurance or software engineering objectives (eq1).

(eq.1)

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

$$Prj = \{ RG_1, RG_2, RG_3, \ldots RG_n \}$$

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

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 (eq2.)

$$W(RG_1)+W(RG_2)+ \ldots + W(RG_n) = 1$$

(eq.2)

$$\sum_{i=1}^{n} RG(W_i) = 1$$

where 1 = Project Total Weight

Each RG is composed out of single requirements (SR) which are also weighted (SRW). The weights on the SR are given in a scale from 1 to 10, with 1 indicating the low complexity and criticality of the requirements while 10 indicates high. The sum of the SRW is irrelevant to the total project weight but relevant to the completion status of the RGW (eq3).

(eq.3)

$$\sum_{i=1}^{n} (R_i W_j)$$

Where:

$i = SR(i), \quad i = \{1 \ldots n\}$

$j = SR(w), \quad j = \{1 \ldots 10\}$

w = weight

$$\sum_{i=1}^{n} (SR_i W_j) \neq R_i W_j$$

If for example a RG has 10 SRs, where most of them have low SRW, then the implementation of the low SRs in weight does not equal the near completion of the RG, if there is one or more SR with SRW more than the sum of the rest of the SRs. (eq4).

(eq.4)

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

Each RG and SR is measured against time and phase of the project development methodology. The phase of the development process form the horizontal axis in a two dimension requirements, management matrix, while the requirements (single and grouped) for the vertical axis.
The MBA-SPI Matrix layout as presented in figure 2, places the RGs and the SRs composing each RG against

the project development phases (DP) and the time associated to each phase.

Taking the MBA-SPI in further down analysis, the development phases are weighted as well. This development phase weight (DPW) comes to be integrated with the Requirement Group Weight (RGW) and Single Requirements Weight (SRW) in order to give a precise picture not only for each RG but of each SR as well.

A development phase (DP) has a weight based on the complexity and the criticality of the phase on the project implementation process. If the testing phase is considered very critical for a specific project for a number of reasons, then this phase will be high weighted than others.
Likewise if the requirements are quite clear in a project, then the analysis of them might have less weight that other phases since the requirements could be self explanatory or the analysis might be integrated in the design phases which in this case, the design phase will get a high weight after integrating the two phases in one. The sum of each DPW equals to the total effort and complexity of the project (eq5)

$$\text{Prj. Phases} = \{ DP_1 + DP_2 + DP_3 + \ldots + DP_n \}$$

$$\text{(eq.5)} \quad \text{Prj. Effort} = \sum_{i=1}^{n} (DP_i W_j)$$

Where:

$i = DP(i), \quad i = \{1 \ldots n\}$

$j = DP(w), \quad j = \{1 \ldots 100\}$

$w = \text{weight}$

The MBA-SPI performs metrics on time intervals based on the project duration, criticality, quality objectives and management needs.
The models can provide quite precise results if the measurements are obtained at least frequently (at least once a week, if not twice).

Analyzing the data gathered in the MBA-SPI matrix the model can provide at any instance: a) precise information on the project status at a given time (eq 6),

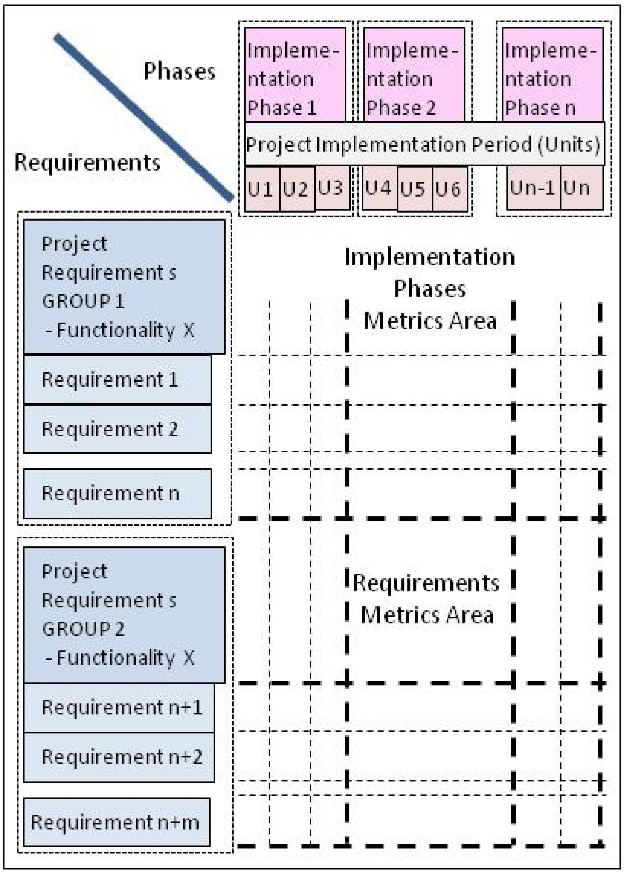

Figure 2. MBA-SPI Matrix Layout.

$$\text{(eq.6)} \quad 100 - \left[\sum_{i=1}^{n} (RG_i) * \left[\sum_{i=1}^{n} \left[\left(\sum_{i=1}^{n} (DP_i W_j) \right) + \left(\frac{\sum_{i=1}^{n} (SR_i W_j) * \left(\frac{Comp_i}{10} \right)}{\sum_{i=1}^{n} (SR_i W_j)} \right) * \left(\frac{RG_i W_j}{10} \right) \right] - RG_i W_j \right] * (-1)$$

b) progress per requirement group over the time (eq7),

(eq.7)
$$\sum_{i=1}^{n}(RG_i) * \left[\sum_{i=1}^{n}\left[\left(\sum_{i=1}^{n}(DP_iW_j)\right) + \left(\frac{\sum_{i=1}^{n}(SR_iW_j) * \left(\frac{Comp_i}{10}\right)}{\sum_{i=1}^{n}(SR_iW_j)}\right) * \left(\frac{RG_iW_j}{10}\right)\right] - RG_iW_j\right]$$

c) progress of specific requirements over the time (eq8),

(eq.8)
$$\sum_{i=1}^{n}(RG_i) * \left[\sum_{i=1}^{n}\left[\left(\sum_{i=1}^{n}(DP_iW_j)\right) + \left(\frac{\sum_{i=1}^{n}(SR_iW_j) * \left(\frac{Comp_i}{10}\right)}{\sum_{i=1}^{n}(SR_iW_j)}\right) * \left(\frac{RG_iW_j}{10}\right)\right] - SR_iW_j\right]$$

d) Phase completion over the time (eq9),

(eq.9)
$$\sum_{i=1}^{n}(RG_i) * \left[\frac{\sum_{i=1}^{n}SR(Comput_i)}{\sum_{i=1}^{n}(SR_iW_j)} * \left(\frac{RG_iW_j}{10}\right)\right]$$

e) deviations from the phase completions time (eq10),

(eq.10)
$$100 - \left[\sum_{i=1}^{n}(RG_i) * \left(\frac{\sum_{i=1}^{n}SR(Comput_i)}{\sum_{i=1}^{n}(SR_iW_j)} * \left(\frac{RG_iW_j}{10}\right)\right)\right]$$

f) Deviation from the requirement completion in a given phase (eq11),

(eq.11)
$$\left((SR_iW_j) * 10\right) - \sum^{n} SR_i(Phase\_Measurement_j)$$

g) Impact of a requirement deviation in the implementation phase (eq12),

(eq.12)
$$\sum_{j=1}^{n}\left(SR_i(Phase\_Measurement_j)\right) - 100$$

h) Impact of the requirement deviation in the overall project (eq13)

(eq.13)
$$\sum_{i=1}^{n}(DP_iW_j) + \left[\sum_{j=1}^{n}\left(SR_i(Phase\_Measurement_j)\right) - 100\right]$$

and other metrics that can form specific metrics categories.

The MBA-SPI model can also be considered as project tracking data mining model, capable to drill down and up on the requirements and also left and right on the project implementation phases and time.
The model can be considered quite accurate since it combines each project tracking element supported with weights correlated in a realistic project tracking matrix giving qualitative progress indicators per requirement per week per project goal.

5 The MBA-SPI Model Tracking Process.

The MBA-SPI model has been designed based on a cumulative computation algorithm on which the inspection readings of each reading period, week or other reading period, are integrated in the progress report of the project, the requirements group progress or the progress of each single requirement.

This computational model is based on the quality and correctness of the data entered.

The readings are performed by project management experts in project tracking, with the support of domain experts. The project manager tracks the progress of each requirement. Each requirement has a maturity level in the time frame that corresponds in a specific project implementation phase. If a requirement (i) needs to be implemented in a given phase (j), with duration (k) weeks, that means that (k) different readings (r), will take place if the reading periods are per week. Therefore the sum of the three readings need to be less or equal to 1 (eq 14) which is the absolute completion status of the requirement in the specific phase and specific implementation time period.

(eq.14)
$$\text{Requirements Implementation Reading } (k) = \sum_{k=1}^{n} R_k(SR_j) \leq 1$$

The risk on this reading process is solely based on the maturity and expertise of the reading team. If for example, a first reading indicated that the requirement in the first week of its implementation on a specific phase is completed by 20% or 50%, then this reading will be recorded and will affect the other reading that will follow, since the total completion of the requirement cannot exceed the 100%. In the same context, the life period that has been given to the requirements can be practically represented in the tracking evolution process. A requirement starts will 0% completion or 100% work to be done (completion^{-1}). As the readings pass the reading time intervals one after the other, the requirement life will eventually come to end when it will reach 0 after the completion of the last project reading or greater than 0 is there has been a phase from which the requirements did not successfully pass. (Eq 15)

(eq.15)
$$\sum_{i=1}^{n}(DP_iW_j) + \left[\sum_{j=1}^{n}\left(SR_i(Phase\_Measurement_j)\right) - 100\right]$$

The whole risk in the effectiveness of the models is the human factor, but since that can be predictable, project tracking teams need to be staffed with the right people that can understand and use scientific methods and approaches towards reaching a goal.

6 Metrics Derived from the MBA-SPI Model.

The depth of the analysis in the MBA-SPI model, along with the massive information gathered in the model's databases, support the development of critical success factors metrics form both the engineering and the business management perspective. The following metrics can be considered as the most valuable ones.

1. Requirements lifecycle completion : Measure the implementation of the requirement in the systems development phases.
2. Requirements group lifecycle completion: Same as the requirements lifecycle completion but for a homogeneous set of requirements operation-wise.
3. Requirements and requirements group completion per implementation phase.
4. Weight of completion per requirements or requirements group against the project target.
5. Weight of completion per requirements or requirements group against an implementation phase.
6. Deviations of the requirements and requirements groups form the project target.
7. Deviations of the requirements and requirements group form the implementation phase target.
8. Impact from a requirement or requirements group deviation on the project target.
9. Impact from a requirement or requirements group deviation on an implementation phase.
10. Impact from the deviation of requirements on the quality of the system being developed.
11. Defect prevention based on the implementation behavior of the requirements or requirement groups.
12. Requirements and requirement group completion rates (fast or slow).
13. Analysis of requirements or requirements group implementation behavior.
14. Weight analysis of implementation phase against implementation time.
15. Weight analysis of requirements or requirements group against project implementation phases.
16. Weight analysis of requirements or requirements group against project implementation time.
17. Project weight management in respect to implementation phases, requirement and requirements group weights.
18. Project maintainability impact based on the requirements and requirements group implementation rates.
19. Software quality impact based on the requirements and requirements group implementation rates.

20. Project success based on requirements and requirements implementation rates.

It must be noted that the metrics that can be derived from the MBA-SPI model cannot be restricted or limited on the ones presented. The interpretation of the massive data generated from the MBA-SPI model allow any engineer or project manager to define new simple or advanced project tracking metrics based on the measurement dimension required to be studied per case and per instance.

7 An Interpretation of the MBA-SPI Model Metrics.

Studying the metrics derived from the MBA-SPI model is easy to imagine the number of other metrics that can be derived from each one. The MBA metrics can be considered as domino metrics.

Taking for instance the Project Status metric which is one metric generated almost form all the other metrics, it can be easy interpreted in both engineering and business terms [16], giving other metrics with engineering or business dimensions.

Engineering wise the metric stands for the project progress, while business wise the metric can sand for the elapse time to operation. The MBA-SPI metrics can form a number of different metric categories according to the interpretation given by the project manager, project owner or any other.

The most significant metrics categories can be the following:

(i) Requirements metrics, indicate the progress of each requirement in the implementation process. Requirements that are left behind schedule or requirements that can be completed at once, document and justify the contradiction of good or bad requirements as well as good or bad requirements elicitation process followed. Bad requirement can be considered as cost centers and need to be either dropped from the project or to be handled with care in order not to generate more requirements due to their interpretation fuzziness.

(ii) Implementation process metrics, indicate the overall implementation process of the project. A figure that can probably indicate that a project is completed by 70% has a huge analysis on each single project requirement in order to justify the 70% and not the 69.5% or any other number.

(iii) Deviation metrics, are usually used to document the distance between the reading and the analysis from the actual and the estimated values.

(iv) Impact metrics, are similar to the deviation metrics since they document the project deviation factors of the project.

(v) Quality metrics, are used towards controlling the project management and project tracking process aligned with the development methodologies and technologies used for the development of the project.

Just like the metrics derived from the MBA-SPI models, the categories cannot be restricted to ones referred as well. It is very likely the same metrics to form two or three different categories depending on the point of view being read or conceived. After all, this is the beauty of the mathematical and statistical sciences.

8 Pre and Post Conditions for Appling the MBA-SPI Model

Every process model, including metric models, is being surrounded by pre and post execution conditions. Pre conditions are based on the maturity of the organization using, in this case, the MBA-SPI model in order to prepare the organizational, management and project requirements, where the model will base its usage and application. Post conditions are considered the maturity requirements towards understanding the model measurements and performing the proper actions or decisions as expected from the measurements predictions.

Some of the MBA-SPI pre conditions towards its applications can be considered the following:

i) a well defined set of project requirements,
ii) fairly requirements stability,
iii) well defined project schedule,
iv) a structured or agile development methodology with defined implementation processes (not necessarily time wise),
v) fairly stability of the project schedule,
vi) availability and capability of a domain expert to define and manage the requirements weights,
vii) availability if a software engineer to define and manage the development process requirements time wise,
viii) availability of a project manager capable to perform project tracking by metrics,
ix) commitment of the project developer to collaborate / participate in the metrics process,

x) commitment of the project owner to participate in the measurements analysis in order to take management actions that will affect the project measurements and project goals and objectives.

Figure 3 presents the staffing requirements towards the successful management of the MBA-SPI execution pre and post conditions.

The MBA-SPI post conditions on the other hand are restricted to the management level of the project steering committee, responsible for the interpretation of the project outcome and the documentation of the expertise gained from the project management process via the MBA-SPI model. Many results will be transferred to the project maintenance contract as risk management factors after being identified by the project implementation process and the behavior of the implementation teams as risks that can affect the quality and the integrity of the delivered project.

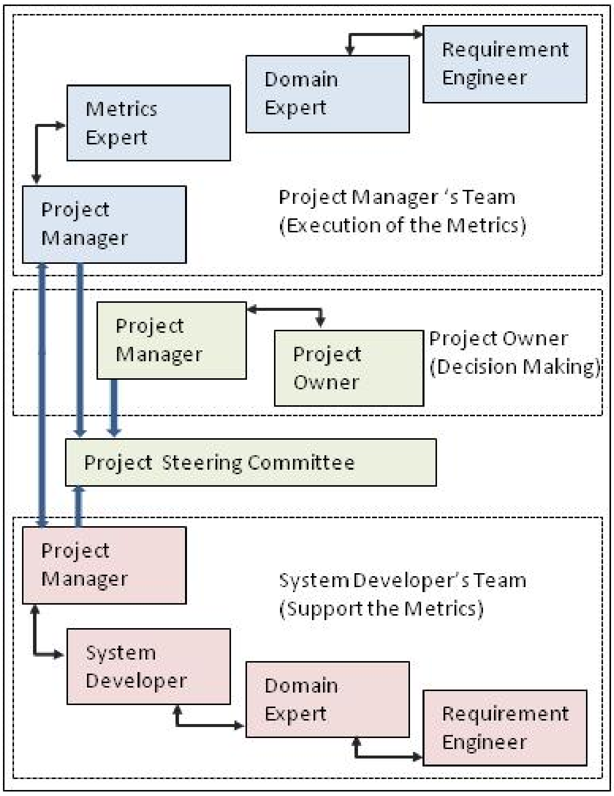

Figure 3. MBA-SPI staffing requirements

9 Further Research

The MBA-SPI model is a quantitative and qualitative weighted tracking model based solely on the project requirements and the project implementation methodology. The integration of the MBA-SPI model with other tracking techniques is an area of further research.

What has been identified after applying the MBA-SPI model in actual and large scale projects, is that the same model can be evolved into an integrated project tracking model incorporating defect prevention theories and techniques.

The model has already been modified to incorporate defect prevention techniques that will be integrated with other defect and risk prevention models like the Raleigh models [17], [18] (eq.16.). This evolution has already being applied in the MBA-SPI model on an attempt to develop a multidimensional project tracking and quality assurance metric system based on the project requirements.

(eq.16)

$$CDF: \; F(f) = 1 - e^{-(t/c)^m}$$

$$PDF: \; f(t) = \frac{m}{t} \left(\frac{m}{t} \right)^m * e^{-(t/c)^m}$$

Where : CDF = Cumulative Distribution Function

PDF = Probability Density Function

If it is possible to track the progress of the project it is then possible to track the changes or the requirements [19].

Requirements changes can generate implementation risks that can affect the quality, cost, time and success of the project implementation process (eq17), (eq 18) [20], [21] [22].

(eq.17)

$$PDE_i = \frac{N_i}{\sum_{m=1}^{i} N_m - \sum_{m=1}^{i-1} N_m}$$

Where :

PDE = Phase Defect Removal Effectiveness

(eq.18)

$$DRE = \frac{\sum_{i=1}^{K-1} N_i}{N}$$

Where :

DRE = Overall Defect Removal Effectiveness of the Development Process

Since the late 70s, [23], [24] heavy research has been conducted on defect prevention based on the requirements behavior not in only in the software development process but also in the software development phases and methods. As result of this research, many models [25], [26] have been developed for software reliability and maintainability based on defect prevention. Raleigh Models developed the infrastructure of defect prevention which can be used further more towards measuring the quality and progress of a project at any instance of its development.

The MBA-SPI model can provide this information and integrate project tracking and project management methods and practices with defect prevention methods, since a defect affects the project implementation schedule and the quality of the delivered software. The information provided by the MBA-SPI model can be considered complete, accurate and reliable and therefore its evolution into an integrated project tracking and defect prevention models can start and actually has already started.

10 Conclusion

The MBA-SPI model conceived and developed by the 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. The MBA-SPI model is a practical model, with scientific background and methodologies, but also simple to apply it successfully if the pre and post conditions of the model are, or nearly in place. Metrics and measurement is an endless process, controlled and evolved by the results of previous initiatives.

The MBA-SPI models is a contribution to the software project management community that can trigger the development of more complex, accurate but primarily practical metric models, incorporating the scientific contribution available, and the success expectations needed to define project reliability and success.

References:

[1] Rai A., Song H., and Troutt M., 'Software Quality Assurance: An analytical survey and research prioritization', Systems Software Vol.40, pp 67-83, 1998

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

[3] 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

[4] ISO/IEC 12207:1995, 'Information Technology-Software life cycle processes – Amendment 1', 2002

[5] Reel, J.S,'Critical Success Factors in Software Projects',IEEE Software, May 1999,pp.106-113

[6] Markopoulos E., Panayiotopoulos J-C., 'A Project Management Methodology Selection Approach based on Practical Project and Organizational Constraints', WSEAS Transactions on Computers, Issue 8, Vol 4, pp 934-942, August 2005

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

[8] Royce W., 'Software Project Management', Addison Wesley, 1998

[9] Mills E., 'Software Metrics', SEI Curriculum Module SEI-CM-12-1.1, December 1988

[10] Hall, E.M., 'Managing Risk Methods for Software Development Risks', Addison-Wesley, 1988

[11] 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

[12] Humphrey W., 'Introduction to the Team Software Process', Addison Wesley, 2000

[13] Holttzblatt k., and Carmel E., 'Requirements Gathering: The human factor', a special issue of ACM, vol. 38, no. 5, May 1995

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

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

[16] The Value of Project Management, Centre for Business Practices Research Report, 2001

[17] Norden P. V., 'Useful Tools for Project Management ', Operations Research in Research and Development, New York, John Wiley & Sons, 1963.

[18] Tobias P., & D. Trindade, 'Applied Reliability, New York, Van Nostrand Reinhold 1986

[19] Putnam L.H., 'A General Empirical Solution to the Macro Software Sizing and Estimating Problem', IEEE Transactions on Software Engineering. Vol. SE-4, 1978, pp345-361

[20] Daskalantonakis, M. K. 'A Practical View of Software Measurement and Implementation Experiences with Motorola', IEEE Transactions on Software Engineering, Vol SE-18, 1992, pp.998-1010

[21] Fagan M. E., 'Design and Code Inspections to Reduce Errors in Program Development', IBM Systems Journal, vol 15, No 3, 1976, pp 182-211

[22] Remus H., S. Zilles, 'Prediction and Management of Program Quality', Proceedings of the fourth international conference on software engineering, Munich IEEE Computer Society, 1979, pp 341-350
[23] Yamada S., M. Ohba, S. Osaki, 'S-Shaped Reliability Growth Modeling for Software Error Detection', IEEE Transactions on Reliability, Vol. R-32, 1983, pp 475-478
[24] Musa J. D., K. Okomuto, 'A Logarithmic Pisson Execution Time Model for Software Reliability Modeling', Orlando, IEEE Computer Society Press, Los Alamitos Calif. 1983, pp. 230-238