Project Management Process Framework for Developing and IT Systems

EVANGELOS MARKOPOULOS
Department of Informatics
University of Piraeus
80 Karaoli & Dimitriou Str., Piraeus
GREECE

JOHN-CHRIS PANAYIOTOPOULOS
Department of Informatics
University of Piraeus
80 Karaoli & Demetriou Str., Piraeus
GREECE

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

CHARALAMBOS MAKATSORIS
School of Engineering and Design
BRUNEL University
Uxbridge, Middlesex UB8 3PH, London,
UNITED KINGDOM

GEORGIOS SAMARAS
Department of Computer Science
University of Cyprus
75 Kalipoleos Av., Nicosia
CYPRUS

TODOR STOILOV
Dept Hierarchical Systems, Institute of Computer and Communication Systems
Bulgarian Academy of Sciences
Acad. Bonchev str. Sofia
BULGARIA

Abstract:- Information Systems Development is a process that needs not only technical capabilities but management capabilities as well. Managing successfully the development process creates a number of qualitative and quantitative benefits to all being involved (suppliers and customers). On the other hand, managing technical people can be both risky and difficult. The methodology that needs to be followed for successful project management need to be quite flexible, measurable, well documented and well defined. This paper takes into consideration the prime stages and processes of 43 methodologies used mainly for information technology project management and creates a methodological framework for managing the systems development process. The framework is supported by best practices, and has been designed in such a way in order to integrate managerial and engineering principles in an adjustable manner. The structure of this framework allows its implementation to support projects of any type, size, and complexity, managed from any perspective (supplied or customer).

Key-Words: - Systems Development, Project Management, Methodologies, Process Framework

1. Introduction

Over the last 40 years many information systems development models have been introduced, with each one of them approaching different development dimensions and strategies. From the all time classic waterfall model [1], to the extreme programming[2] and the AGILE methods[3], systems and software development has been evolved significantly covering all types of information systems and development goals.

Despite the fact that new systems development techniques have been evolved based on system parameterization, system reuse and system integration, [4],[5].[6], most of the systems being build prefer the custom-made approach[press], with the human interaction factor in a very critical role.

On the other hand, the progress in system development methods and frameworks created a very matured engineering infrastructure, which unfortunately, due to custom systems development preferences, seems not to be enough in order to achieve qualitative results.

Developing successful information systems depends on many management perspectives, practices and approaches that will support the engineering goals[7],[8].

2. The Integration of management and engineering practices need

Managing technical people, and managing IT and software in particular customers, can be more difficult than managing the development process itself. Project management, quality assurance and human resource management are disciplines that need to be integrated with the engineering ones [9], [10], in order to perform qualitative and quantitative systems development.

This disciplines integration is not a new concept. SEI's SW-CMM [11], was one of the first models that differentiated the management (maturity level 2) from the engineering practices maturity level 3) but still included them in the same model. On the other hand the degree on project management process integration into the engineering systems development models can differ significantly among the management or engineering goals that have been placed in each project specifically. Nevertheless in order to overcome the old and new software crisis[12],[13] in which we are into, inter-discipline process models need to be introduced and evolved.

3. A Project Management Framework for Systems Development

An Information Systems Development Management Framework (INSYDEMF) has been developed by taking into consideration the practical need for management and engineering practices integration.

In order to develop INSYDEMF, a detailed analysis of 43 project managing methodologies, guidelines, and frameworks, has been conducted towards the identification of the commonly used practices in system development, software engineering and project management under different goals and dimensions.

INSYDEMF in a framework, not a methodology. IN INSYDEMF, a systems development life cycle forms the basis of the project management framework that can be used for managing projects in organizations that have the capability or the responsibility to develop information technology systems. The goal of INSYDEMF is to be used mainly be the software intensive small to medium size enterprises which have the most difficulties to launce software process improvement programs and structured project management efforts[14].

INSYDEMF's room for improvement depends on the maturity of the organization that will be using it and on the complexity of the projects that comes to support.

4. Phases in the Information Systems Development Management Framework

The structure of INSYDEMF is composed of fifteen (15) phases as show in table 1. The first column indicates the generic name of each phase, the second column indicates the methodologies supporting this generic name, and the third column indicates other methodologies that support it, but with slight naming and activities deviations.

Phase Name	Supportive Methodologies	Related Supporting Methodologies
Analysis	SDLC, SEFER, DoD2167^{A}, IE	LCM-AIS, Princeton, Ariadne-SD
Implementation Planning	SDLC, ITIL, ITPM, SW-CMM, CBAM, IE	DoD2168, ISO9000-3, LFA, Ariadne-PM, SE-CMM
Design	SDLC, AIM, PROMPT, ASAP, ISO9000-3, SEFER, IE, DoD2167^{A}	ITPM, AUSGuidelines, Princeton, Ariadne-SD
Development	SDLC, LCM-AIS, AIM, PROMPT, ASAP, ITPC, ISO9000-3, SEFER, IE, DSDM	IDEAL, SE-CMM, DoD2167^{A}, Princeton, Ariadne-SD
Integration.	SCALABLE	ITPM, SE-CMM, DoD2167^{A}, Ariadne-SD
Installation	PROMPT, ISO9000-3	Princeton, Ariadne-SD
Testing	AIM, SEFER, DoD2167^{A}, DSDM	SE-CMM, ISO9000-3, Ariadne-SD
Operation	PROMPT, LCM-AIS, ASAP, ITPC	Ariadne-SD
Documentation	Ariadne-SD, AIS	SUPRA, TENSTEP, AIM, ISO9000-3
Acceptance	ISO9000-3	Ariadne-SD
Maintenance	ISO9000-3, SEFER	Ariadne-SD
Change management.	WWPMM, ISO900-3, DSDM, AIS	Ariadne-PM
Risk management	WWPMM, IPM, SCALABLE, TENSTEP, EUROMETHOD, ITPM, SE-CMM,	CBAM, AUSGuidelines, Ariadne-PM
Project team management (HRM)	WWPMM, ITPM, ISO9000-3, DSDM	SCALABLE
Software project tracking and oversight	WWPMM, SW-CMM,	SUPRA, SDLC, ITPM, DoD2167^{A}, DoD2168, ISO9000-3, CBAM, Ariadne-PM, AUSGuidelines, SE-CMM

Table 1. Phases composing the Project Management Framework for System Development

The methodologies listed in the second column support the phases listed in the first column of the table in a very precise way. Most of the phase activities are quite relevant between the phase definitions and goals in all methods.

Unlike the second column of table 1, the methodologies listed in the third column support conceptually the related phases but from a different perception. Fr example, the activities of the 'design' phase, for example, can be more of less found in phases of other related methodologies not referred as 'Design', but as 'Detailed Design', Preliminary Design', 'Project Design', 'Design Phase', or even 'Application and Technical Architecture'.

All these different versions of the 'Design' phase have the same goal and scope of what a Design phase covers in both activities and deliverables. In a similar way, the 'Development' phase has been referred in the supporting methodologies as 'Coding and Testing', 'Development Stage', 'Parameterization', or ever 'Evolve System Architecture'. This set of definitions cover all the dimensions of systems development including 'coding' for custom software development, and 'parameterization', for the implementation of ERP's, WMS, MIS, or other commercial software systems.

Taking further this analysis down to more managerial tasks like the 'Project Tracking' it has been found that this definition has been called in supporting methodologies as 'Project monitoring and control', 'Reviews and inspections', 'Monitoring Estimates', 'Progress reporting' or 'Project Review Report'. All of these phases support the concept of project tracking through a different dimension but with a common logical denominator, since project tracking is actually a set of activities that document the progress of a project, using inspections, reviews and walkthrough to identify consistency with the planned and estimated implementation scheduled values.

5. The Requirements Concept

INSYDEF is a framework with all its phases based on the requirements concept. Requirements elicitation, validation and analysis activities usually take place before the implementation of the project [15], [16]. The requirements can be developed by either other groups of people (non systems developers), such as the business analysts, domain experts or business experts, in collaboration with the customers, government regulations, corporate goals or the market trends , [17].

INSYDEF takes the requirements as the backbone of its operation and management innovation. By managing the evolution of the requirements from plain text, to analysis diagrams (use cases, etc), to design diagrams (activity, stage, deployment diagrams, etc), to code, to test cases, to documentation paragraphs, to acceptance scenarios, to operation indexes, etc, qualitative project management can be obtained [18].

The requirement can be perceived as a living organism, or as reference point to every engineering activity that supports any development life cycle under any development model. In INSYDEF, the requirement concept is also the one that enables tracking, measurement, and HRM activities to enter into the so called 'engineering practices', transforming this way the system development methodology, into a systems development framework.

6. System Development Framework description and dependencies

INSIDEMF's structure has been developed under the waterfall model concept. After all, the waterfall model has been the basic inspiration of all the other development models that followed [19], [20], [21], with slight alterations in its interpretation.

Table 2 describes the phases of the Project Management Framework for System Development, while indicating the dependencies among them.

Phase	Description	Dependent Phase	Dependency Type
Analysis	Analysis of the project scope, environment, technologies and preliminary requirements.	Project Planning	Start>Finish or Finish>Start
Project Planning	Project decomposition, planning, estimating and scheduling on resources, effort, and budget.	Analysis	Start>Finish or Finish>Start
Design	Technical interpretation of the project requirements with design tools and technologies.	Project Planning or Analysis	Finish>Start
Development	Realization of the project requirements with programming languages, reusable systems or parameters setting.	Design	Finish>Start
Integration.	Unification of all systems being developed, as well as adjacent systems to complete the project.	Development	Finish>Start
Installation	Initial delivery of the system in test environment	Development Integration	Partial Finish>Start Finish>Start
Testing	Structured system validation and verification based on the requirements and the documentation derived up to this phase.	Development Installation	Finish>Start Finish>Start
Operation	Systems performance and functionality measurement and maintenance under a given time frame and conditions.	Testing	Finish>Start
Documentation	Final integration of the documents developed in the framework phases.	Testing Operation	Partial Finish>Start Finish>Start
Acceptance	System acceptance. End of project.	Operations Documentation	Finish>Start Finish>Start
Maintenance	Repetition of the framework in a less intensive way at any system modification.	All Engineering Phases	Finish>Start
Change management	Documentation, implementation, scheduling, monitoring and impact analysis of systems changes.	All Engineering Phases	Start>Start
Risk management	Identification of technical and non-technical risks in systems implementation and maintenance.	All Engineering Phases	Start>Start
Project team management	Managing technical people (Human Resource Allocation)	All Engineering Phases	Start>Start
SW project tracking t	Monitoring the development process based on inspection and reviews.	All Engineering Phases	Start>Start

Table 2. Phase Dependencies of the Project Management Framework for System Development

From the dependencies of the phases included in this project management methodological framework it is clearly identified that a number of pure management activities have been included in it. Project tracking for example, or change management are phases with activities that can be applied in all engineering phases. On the other hand it is also clear that this methodological framework is based on the engineering discipline covering all major development phases. Figure 1 presents more clear, the relationships and dependencies of the phases composing this framework.

7. System Development Framework Milestones and Prime Deliverables

Every phase is composed by a set of activities. An activity is an action and an action produces a deliverable that justifies its implementation [22].

The project management framework for system development has also a minimum set milestones whose implementation indicates the framework's proper usage.

In a similar way, basic sets of documents derived from the framework activities composing the project management documentation. This set of prime documents is the minimum documentation of the framework.

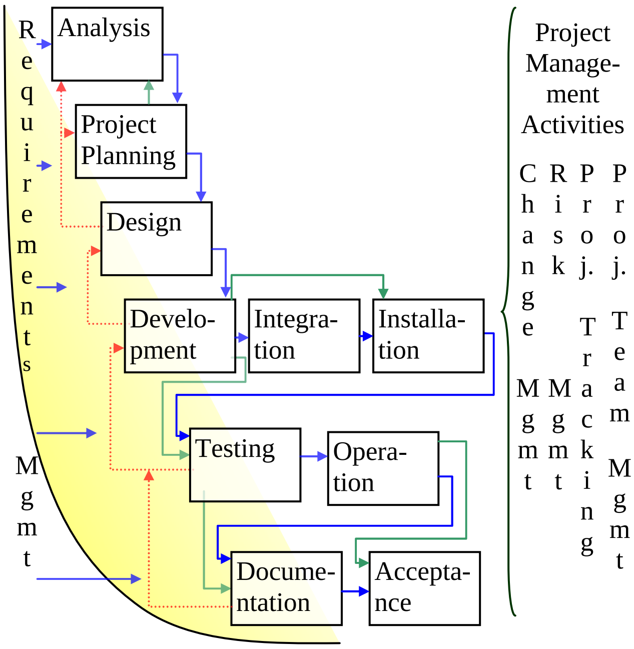

Figure 1. The Information Technology Systems Development Management Framework Model

Secondary (additional) documents can be created based on the documentation of each phase activities, in order to achieve a more accurate management picture, quality assurance and technology re-use.

Table 3 lists the major milestones and deliverables of the framework.

Phase	Milestone	Prime Deliverables (Documents)
Analysis	▪ Analysis of the project scope environment, technologies and preliminary requirements.	▪ Analysis Document.
Project Planning	▪ Decomposition Acceptance. ▪ Estimations Acceptance.	▪ Project Plan.
Design	▪ Design Acceptance. ▪ Design compliance with analysis.	▪ Physical&Logical Design Docum ▪ Database Design Docum.
Development	▪ Code/functionality compliance with the design. ▪ Unit test.	▪ Data dictionary
Integration.	▪ Systems components collection. ▪ System Build. ▪ Integration test	▪ Integration plan & results. ▪ Integration test plan.
Installation	▪ Customization on real working environment. ▪ System Build.	▪ Installation Plan & Results.
Testing	▪ Test scenario collection. ▪ Test environment set-up.	▪ Test Plan. ▪ Test Results.
Operation	▪ User Training.	▪ Defect Log.
Documentation	▪ Update documents. ▪ Complete contractual documents.	▪ Updated Version of Framework documents. ▪ Contractual Documents.
Acceptance	▪ Acceptance Test.	▪ Acceptance test plan
Maintenance	▪ Planning and Estimations. ▪ Testing.	▪ Update Related Documents.
Change management.	▪ Accept change. ▪ Test change implementat.	▪ Change docum log.
Risk management	▪ Risk Acceptance. ▪ Risk Plan. ▪ Risk Implementation.	▪ Risk Document ▪ Risk implem. document.
Project team management	▪ HR team definition ▪ HR management model	▪ HR management model.
Software project tracking and oversight	▪ Tracking Plan ▪ Implementation of Reviews and Inspections	▪ Tracking Plan Document. ▪ Progress Reports (in defined frequency).

Table 3. Phase Milestones and Deliverables of the Project Management Framework for System Development

A close correlation between the phase's milestones and documentation can be perceptible at a glance in table 3. This milestone-deliverable relationship is due to the fact that in order for a milestone to be reached, many activities have been implemented, whose result is documented. On the other hand in an attempt to present a framework that will not add large administration overhead to the management process, the framework documents are related in an almost corresponding way to the milestones. Figure 2 presents the number of milestones and documents produced in every phase of the framework.

The phases of 'Change Management', 'Project Tracking' and 'Operation' do not produce actual documentation but a number of reports, whose sum produces at the end of the project the specific phase documentation. In a similar way the phases of 'Documentation', and 'Maintenance' also do not produce a specific document but an update of the documents developed in the previous implementation phases.

8. Advantages and Disadvantages for Using the INSYDEMF

Every process framework can be easily accused and or accepter for its integrity, structure, usability, and other characteristics. The ease of making judgments derives from many parameters quite different for every judge. Personal process maturity is the most significant factor that makes the go/no-go decision. Unfortunately these judgment factors can be of high risk since if all project managers, business owners, development managers and systems / software engineers had the maturity judge, then the software engineering discipline would have been nearly perfect, with no improvement needs. Unfortunately this is not that case. Introducing state of the art systems development frameworks is very likely never to be used in practice. INSYDEMF is strong enough to be placed in practice at once, and weak enough to provide detailed support to state of the art projects or minds.

Nevertheless, some of the weak points of INSYDEMF could be its ‘old fashion’ waterfall approach, its limitation to the management activities (why not adding contract management, configuration management, performance management, etc), and its limitation on milestones and deliverables definition. On the other hand, some of the benefits could be its simplicity, the requirements management concept, the integration of management and engineering activities and the target group that is being designed to serve.

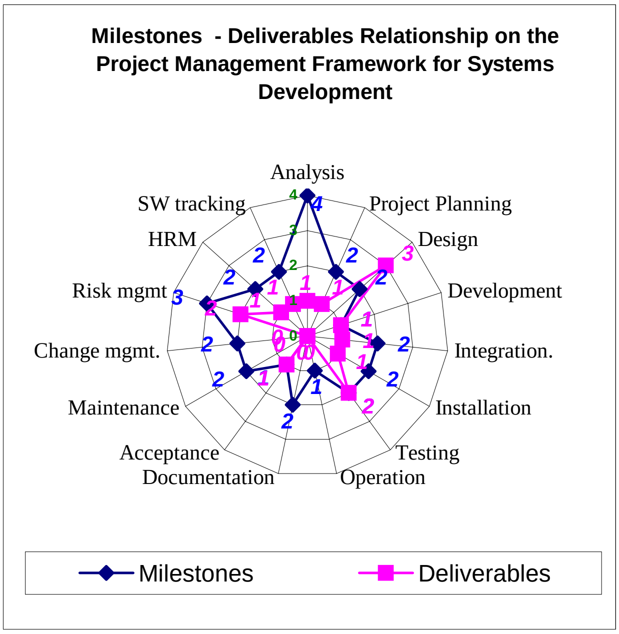

Figure 2. Graphical interpretation of milestone and deliverable correlation on the Project Management Framework for System Development

At any case more weak and strong points could be found, but it seems that what is weak for one instance could be strong for another.

9. Results

The engineering phases used in project management aim to manage the way an information technology project is implemented from the developers point of view.

This paper introduced a practical project management approach towards managing the development process under any perpective. A prime result that derives from this work is based on a process analysis based on many international methodologies. This analysis provides a practical platform of best practices and trends towards managing the systems development process.

Another result is that the methodology has been designed to support the small to medium size enterprises in their development efforts by embedding in the development process project management activities. This project management and software engineering integration gives to the methodology a wider application spectrum. Secondary results derive from the structure of the methodology. The requirements concept, the deliverables, the dependencies and the milestones give an in depth support towards using the framework.

In a nutshell, this work contributes to the systems development effort an adjustable, best practice based, and practical management tool. Unfortunately this model cannot be considered as the best one around, or as a silver bullet. After all, we shall keep in mind that even the SEI-CMM generated much debate, despite the fact that it provides a well defined set of indicators on systems development practices [23],[24].

References:

[1] Royce W. W., ‘Managing the Development of Large Software Systems’, Proceedings of IEEE WESCON, pp 1-9, 1970

[2] Shore J., ‘Continuous Design’, IEEE Software, Vol 21, Issue 1, pp. 20-22, January 2004

[3] Ambler S., ‘Agile Modeling’ John Wiley & Sons 2001

[4] Khan A., ‘Implementing SAP with an ASAP Methodology Focus’, Writers Club Press, 2002

[5] Yourdon E., ‘Sodtware Reuse’, Application Development Strategies, vol. 1, no. 12, December 1994, pp. 1-16

[6] Sharon d., and Bell R., ‘Tools that Bind: Creating Integrated Environments’, IEEE Software, March 1995, pp. 76-85

[7] Pressman, R., and Ince D., ‘Software Engineering. A practitioner’s approach.

European Adaptation' Fifth Edition, McGraw Hill, 2000

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

[9] Humphrey W., 'Managing The Software Process', Addison Wesley, 1989

[10] Altman R., et. all., 'Middleware: The Glue for Modern Applications', Gartner Group, Strategic Analysis Report, July, 1999

[11] Carnegie Mellon University. Software Engineering Institute., 'The Capability Maturity Model. Guidelines for Improving the Software Process', Addison Wesley, 1995

[12] Arthur, L. J. Measuring Programmer Productivity and Software Quality. New York: John Wiley, 1985

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

[14] Markopoulos E., 'An Empirical Adjustable Software Process Assessment Model for Software Intensive Small and Medium Size Enterprises', 7th European Conference on Software Quality, Conference Notes Tammerpaino, Tampere, Finland, pp16-19, 2002

[15] Reifer D., 'Requirements Engineering', in Encyclopedia f Software Engineering, Wiley, 1994, pp. 1043-1054

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

[17] Wyder T., 'Capturing Requirements with Use-Cases', Software Development, February 1996, pp 37-40

[18] Robertson S., Robertson J., Mastering the Requirements Process, Addison Wesley, 1999

[19] Hematpour S., 'Experience with evolutionary prototyping in a large software project', ACM SIGSOFT Software Engineering Notes, Vol.12, no.1, 1987

[20] Boehm B., 'A Spiral Model of Software Development and Enhancement', IEEE Computer, vol. 21, no. 5, May 1988 pp 61-72.

[21] Davis A., and Sitaram P., 'A Concurrent Process Model for Software Development', Software Engineering Notes, ACM Press, vol. 19, no. 2, Aprl 1994, pp. 38-51

[22] Sommerville I., 'Software Engineering', Sixth Edition, Addison Wesley, 2001

[23] Bollinger, T. and McGowen, C., 'A critical look at software capability evaluations', IEEE Software, July 1991, pp 25-41

[24] Gild, T., 'What is level Six', IEEE Software, January 1996, pp. 97-98, 103