ISSN 1512-1232

# THE INFLUENCE OF ROLE AND CRUCIAL ATTRIBUTES OF SOFTWARE DEVELOPMENT TEAM IN THE SOFTWARE PROJECT DEVELOPMENT PROCESS

Dr.S.S.Riaz Ahamed

Principal, Sathak Institute of Technology, Ramanathapuram, Tamilnadu, India.
Email:ssriaz@ieee.org, ssriaz@yahoo.com

***Abstract***
*Software development process is a team activity and it involves different phases engaged with group of persons involved in different activities. Every individual has his own responsibility in completion of his or her process. Though they have given a responsibility, it is essential to have a successful project leader for the smooth execution of the project. He has to apply a problem solving management style to work on the project.*

## 1 INTRODUCTION

An effective software project management focuses on three vital aspects. They are people, problem, and process.

**People:** The people management maturity model defines the following key practice areas for software people, they are recruiting, selection, performance management training, compensation, career development, organization and work design, and team/culture development. Increasingly complex applications can only be developed by helping people to grow, attract, motivate, deploy and retain the talent needed to improve their software development capability.

**The problem aspect:** Project planning starts when the objectives and scope are established, alternate solutions should be considered, and technical and management constraints should be identified. Without this information, it is impossible to define reasonable (and accurate) estimates of cost; an effective assessment of risk; a realistic breakdown of project tasks; or a manageable project schedule that provides a meaningful indication of progress. The software developer and the user interact to define project objectives and scope. In many cases, this activity begins as part of the system engineering process and continues as the first step in software requirement analysis. Objectives identify the overall goals of the project without considering how these goals will be achieved. Scope identifies the primary data, functions, and behaviors that characterize the problem, and more important, attempts to bound these characteristics in a quantitative manner.

If only, the project objectives and scope are understood, alternate solutions are considered to select a ”best” approach, given the constraints imposed by delivery deadlines, budgetary restrictions, personnel availability, technical interfaces, and myriad other factors.

**The Process Aspect:** The process aspect in software provides the framework from which emerges the plan for software development can be established. A small number of framework activities are applicable to all software projects, regardless of their size and complexity. A number of different task sets- tasks, milestones, deliverables and quality assurance points- enable the framework activities to be adapted to the characteristics of the software project and the requirements of the software team. Last but not the least, umbrella activities – such as software quality assurance, software configuration management, and measurement- overlay the process model. Umbrella activities are not dependent on any framework activity, which occur throughout the process.

ISSN 1512-1232

## 2 TEAM

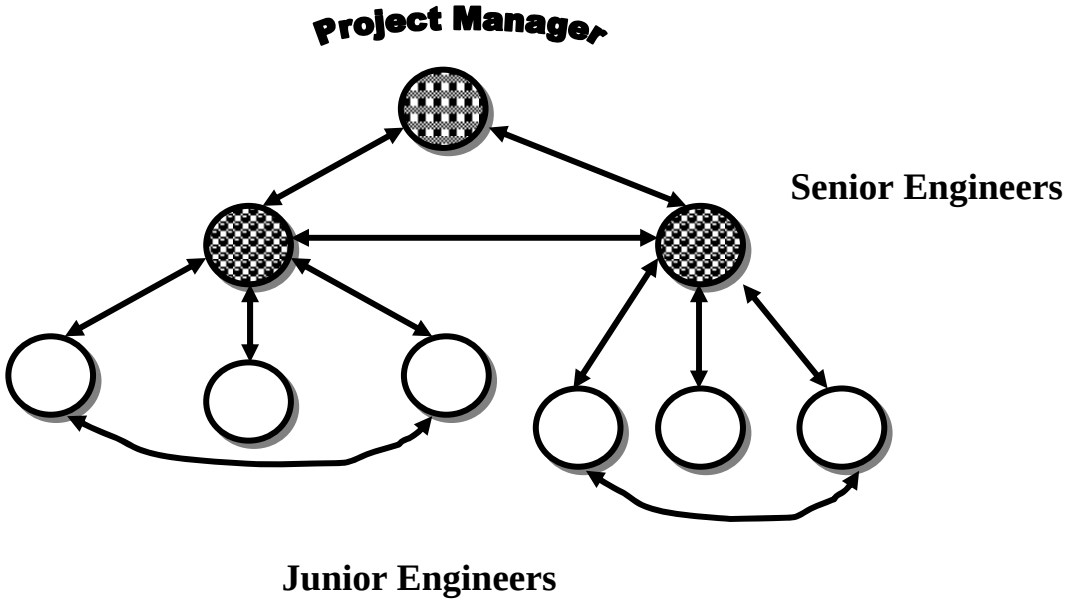
The players in the software arena can be divided into five categories for the purpose of study:

1. Senior managers, who define the business issues and influence the project,.
2. Project (technical) mangers plan, motivate, organize, control and they are responsible for the product.
3. Practitioners, who deliver the technical skills that, are necessary to engineer a product or application.
4. Customers, who specify the requirements for the software to be molded.
5. End users, who utilize the software after it is launched properly.

The team structure found to be good based on the selection scheme by the management. Mantei suggests three best models to device a best project team.

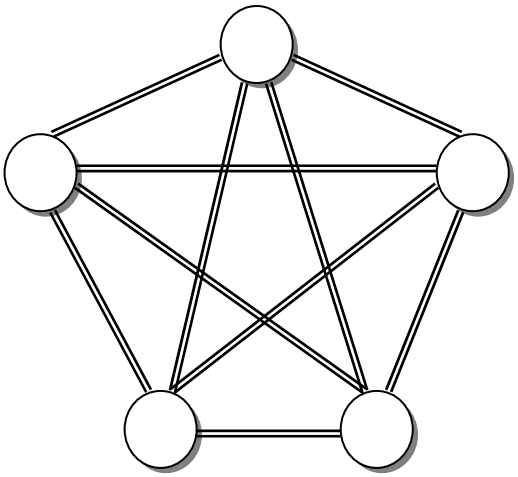
- Democratic Decentralized (dd)
- Controlled decentralized (cd)
- Controlled Centralized (cc)

**Democratic Decentralized:** This engineering team does not have any specific team leader. Rather it appoints "task coordinators" for a period of short duration. The communication is horizontal and the decision is made by the group consensus.

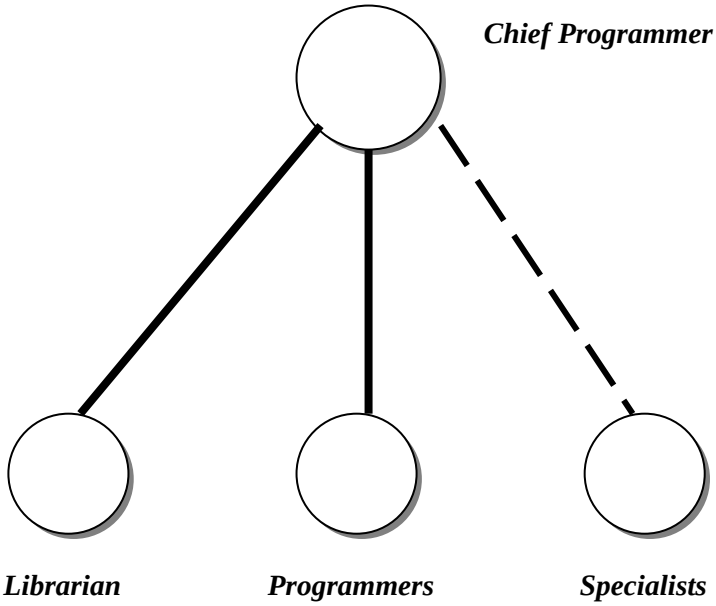



**Controlled decentralized:** In this engineering team, the process is divided into main task and sub task. A permanent leader is been appointed and he takes care of the main task. The next level of authority looks after the subtask. Problem analyzing being the group activity but the decision is made only by the team leader.

ISSN 1512-1232



**Controlled Centralized:** Top-level problem solving and internal team coordination are managed by a team leader.




The team morale is affected by the length of time the team will "live together". DD team structures results in high morale and job satisfaction and are therefore good for long lifetime teams. The DD team structure is best applied to problems with relatively low modularity because of the higher volume of communication that is required.

Mantei also describes seven factors for planning the structure of the engineering teams. They are

- The difficulty of the problem to be solved
- The size of the resultant program(s) in lines of code or function points
- The time the team will stay together (teamlifetime)
- The degree to which the problem can be modularized
- The requirement quality and reliability of the system to be built
- The rigidity of the delivery date
- The degree of sociability (communication) required for the project.

The major that occur in teams and their remedies are given below.

ISSN 1512-1232

| *RISK ITEMS* | *RISK MANAGEMENT TECHNIQUES* |
|---|---|
| Personnel shortfalls | Staffing with top talent; job matching; team-building; key-personnel agreements; cross training; pre-scheduling key people |
| Unrealistic schedules and budgets | Detailed multisource cost & schedule estimation; design to cost; incremental development; software reuse; requirements scrubbing. |
| Developing the wrong software functions | Organization analysis; mission analysis; ops-concept formulation; user surveys; prototyping early user's manual. |
| Developing the wrong interface | Prototyping scenarios; task analysis; user characterization (functionality, style, workload) |
| Gold Plating | Requirements scrubbing; prototyping; cost benefit analysis; design to cost |
| Continuing stream of requirements changes | High change threshold; information hiding; incremental development(defer changes to later increments) |
| Shortfalls in externally furnished components | Benchmarking; inspections; reference checking; compatibility analysis. |
| Shortfalls in externally performed tasks | Reference checking; pre-award audits; award-fee contacts; competitive design or prototyping; teambuilding. |
| Real-time performance shortfalls | Simulation; benchmarking; modeling; prototyping; instrumentation; tuning |
| Straining computer science capabilities | Technical analysis; cost benefit analysis; prototyping; reference checking. |

Constantine suggests four "organizational paradigms" for software engineering teams:

1. A closed paradigm structures a team along a traditional hierarchy of authority. Such teams can work well when producing software that is quite similar to past efforts, but they will be less likely to be innovative when working within the closed paradigm.
2. The random paradigm structures a team loosely and depends on individual initiative of the team members. When innovation or technological break-through is required, teams following the random paradigm will excel. But such teams may struggle when "orderly performance" is required.
3. The open paradigm attempts to structure a team in a manner that achieves some of the controls associated with the closed paradigm but also much the innovation that occurs when using the random paradigm. Work is performed collaboratively with heavy communication and consensus-based decision-making. Open paradigm team structures are well suited to the solution of complex problems, but may not perform as efficiently as other teams.
4. The synchronous paradigm relies on the natural compartmentalization of a problem and organizes team members to work on pieces of the problem with little active communication among themselves.

## 3 LEADERSHIP

A successful leader is one who can motivate persons around him and make them follow him. His work and conduct is a living example for others to emulate. People flock around the leader impressed by certain values or characteristics that the leader has. He is of course a good communicator and understands the problems that beset the work of the persons around him and has concern and regard for them. As he has a good understanding of situations around him, he is able to suggest remedies and solve problems.

Leadership is a process of influence on a group. It is an important part of a manager's job. Effective leadership is necessary for inspiring the people to work for the accomplishment of a given objectives. It provides a cohesive force, which holds the group intact and develops a spirit of

ISSN 1512-1232

cooperation. Effective leadership is essential for efficient direction of human efforts towards the predetermined goals.

Chester Bernard viewed leadership as the quality of behavior of individuals whereby they guide people or their activities in organizing efforts. A leader interprets the objectives of the people working under him and guides them towards the achievement of those objectives. In other words of Louis A.Allen,"A leader is one who guides and directs other people. He gives the efforts of his followers a direction and purpose by influencing their behavior."

Leadership is a process of influencing the subordinates so that they cooperate enthusiastically in the achievement of group goals. According to Theo Haimann, "Leadership is the process by which an executive imaginatively directs, guides and influences the work of others in choosing and attaining specified goals by mediating between the individuals and the organization in such manner that both will obtain maximum satisfaction."

## *Characteristics of Leadership*

An analysis of the above definitions of leadership reveals that it has the following characteristics:

1. Leadership is a process of influence: Leadership is a process whose important ingredient is the influence exercised by the leader on group members. A person is said to have an influence over others when they are willing to carry out his wishes and accept his advice, guidance and direction. Successful leaders are able to influence the behavior, attitudes and beliefs of their followers.
2. Leadership is related to a situation: When we talk of leadership, it is always related to a particular situation, at a given point of time and under specific set of circumstances. That means leadership styles will be different under different circumstances. At one point of time, the subordinates may accept the autocratic behavior of the leader while at a different point of time, and under different situation, only participative leadership style may be successful.
3. Leadership is the function of stimulation: Leadership is the function of motivating people to strive willingly to attain organizational objectives. Leaders are considered successful when they are able to subordinate the individual interests of the employees to the general interests of the organization. A successful leader allows his subordinates to have their individual goals set up by themselves in such a way that they do not conflict with the organizational objective.

The following are some of the crucial attributes of successful software project managers:

1. Hiring skills: Few decisions are as important as hiring decisions. Placing the right person in the right job seems obvious but it is surprisingly hard to achieve.
2. Avoiding adversarial relationships among stake-holders is a prerequisite for success.
3. ***Decision making skill:*** The jillion books written about management have failed to provide a clear definition of this attribute. We all know a good leader when we run into one, and decision making skills seems obvious despite its intangible definition.
4. ***Team-building skill:*** Teamwork requires that a manager establish trust, motivate progress, exploit eccentric prima donnas, transition average people into top performers, eliminate misfits, and consolidate diverse opinions into a team direction.
5. ***Selling skill:*** Successful project managers must sell all stakeholders (including themselves) on decision and priorities, sell candidates on job positions, sell changes to the status quo in the face of resistance, and sell achievements against objectives. In practice, selling requires continuation negotiation, compromise and empathy.

ISSN 1512-1232

## 4 COMMUNICATION

Communication is the key to everything in the universe and People evolve process and they rectify problem through communication and interaction. How effectively this is done is a function of harmony among the three vital forces. Communication is the important factor in the project coordination. The best way of communication makes the information reach the round table in proper way and exact way.

Kraul and Streeter examined a collection of project coordination techniques that are categorized as follows:

**Formal, impersonal approaches:** Include software engineering documents and deliverables (e.g source code), technical, memos, project milestones, schedules and project control tools, changes requests and related documentation, error tracking reports and repository data.

**Formal, interpersonal procedures:** Focus on quality assurance activities applied to software engineering work products. These include status review meetings and design and code inspections.

**Informal, interpersonal procedures:** Include group meetings for information dissemination and problem solving and "collection of requirements and development staff".

**Electronic communication:** Encompasses electronic mail, electronic bulletin boards, Web sites, and by extension, vide-based conferencing systems.

**Interpersonal network:** Informal discussions with those outside the project who may have experience or insight that can assist team members.

**Award Fee Flow down plan**

The implementation of the award fee flow down plan was intended to achieve the following objectives:

- Reward the entire team for excellent project performance.
- Reward different peer groups relative to their overall contribution.
- Substantially reward the top performers in every peer group
- Minimize attrition of good people.

The basic operational concept of the plan:

- Management defined the various peer groups (systems engineering, software engineering, business administration, and administration)
- Every six months, the people within each peer group ranked one another with respect to their contribution of the project. The manager of each peer group also ranked the entire team. The manager compiled the results into global performance ranking of the peer group.
- Each award fee was determined by the customer at certain major milestones. Half of each award fee pool was distributed to project employees.
- The algorithm for distributions to project employees was fairly simple. The general range of additional compensation relative to each employee's salary was about 2% to 10% each year.
- The distribution to ach peer group was made relative to the average salary and total number of people within the group. The differences in employee's salaries within each group

## 5 PERSONALITY TRAITS

The software project leader often concentrates on understanding the problem to be solved, managing the flow of ideas, and at the same time, letting everyone in the team know (by words, and far more important, by actions) that quality alone counts.

ISSN 1512-1232

Four key traits are identified for the team leader for effective management of a software project:

**Problem solving:** An effective software project development-leader, identifies the technical and organizational issues that are most relevant, systematically structure a solution and properly motivate other practitioners to develop the solution, apply lessons learned from past projects to new situations, and remain flexible enough to change if it is discovered that the steps initiated are not fulfilling the objectives.

**Leading Identity:** A leader, who leads his team apart from being confident, should take control when necessary and the assurance to allow good technical people to follow their instincts as long as they are in tune with the objectives.

**Achievement**: Accomplishment and initiative should be rewarded and the leader should often demonstrate "this" through his own actions that controlled risk taking are encouraged and not punished.

**Tact and Team Building:** A leader who has the quality of "reading the team's mind", must be able to understand verbal and non-verbal signals and react to the needs of the people from where it springs from. A leader must remain under control in high-stress environment.

## 6 PROJECT ORGANIZATIONS

The structure of the organization can be tailored to the size and circumstances of the specific project organizations. The main fault of the default organization are as follows:

- The project management team is an active participant, responsible for producing as well as managing. Project management is not a spectator sport.
- The architecture team is responsible for real artifacts and for integration of components, not just for staff functions.
- The development team owns the component construction and maintenance activities. The assessment team is separate from development. This structure fosters an independent quality perspective and focuses a team on testing and product evaluation activities concurrent with on-going development.
- Quality is everyone's job, integrated in all activities and checkpoints. Each team takes responsibility for a different quality perspective.

### 6.1 Software Management Team

Most projects are over constrained. Schedules, costs, functionality, and quality expectations are highly interrelated and require continuous negotiation among multiple stakeholders who have different goals. The software management team carries the burden of delivering win conditions to all stakeholders. In this regard, the software project manager spends every day working about balance

***Life-Cycle Focus***

| **Inception** | **Elaboration** | **Construction** | **Transition** |
|---|---|---|---|
| Elaborating phase | Construction phase planning | Transition phase | Customer satisfaction |
| Planning | Full staff recruitment | planning | Contract closure |
| Team formulation | Risk resolution | Construction plan | Sales support |
| Contract baselining | Product acceptance criteria | Optimization | Next-generation |
| Architecture costs | Construction costs | Risk management | planning |

The software management team takes ownership of all aspects of quality. In particular, it is responsible for attaining and maintaining a balance among these aspects so that overall solution is adequate for all stakeholders and optimal for as many of them as possible.

### 6.2 Software Architecture Team

ISSN 1512-1232

The software architecture team is responsible for the architecture. This responsibility encompasses the engineering necessary to specify a complete bill of materials for the software and engineering necessary to make significant make/buy trade-offs so that all custom components are elaborated to the extent that construction assembly costs are highly predictable. For any project, the skill of the software architecture team is crucial. It provides the framework for facilitating team communications, for achieving system-wide qualities, and for implementing the applications. With a good architecture team, an average development team can succeed. If the architecture is weak, even an expert development team of superstar programmers will probably not succeed.

***Life cycle focus***

| **Inception** | **Elaboration** | **Construction** | **Transition** |
|---|---|---|---|
| Architecture prototyping<br>Make/buy trade-offs<br>Primary scenario definition<br>Architecture evaluation criteria definition | Architecture baselining<br>Primary scenario demonstration<br>Make/buy trade-off baselining | Architecture maintenance<br>Multiple componenet issue Resolution<br>Performance tuning<br>Quality improvements | Architecture maintenance<br>Multiple-component issue resolution<br>Performance tuning<br>Quality improvements |

To succeed the architecture team must include a fairly broad level of experience including the following:

- Domain experience to produce an acceptable design view (architecturally significant elements of the design mode) and use case view (architecturally significant elements of the use case model)
- Software technology experience to produce an acceptable process view (concurrency and control thread relationships among the design, component, and deployment models), component view (structure of the implementation set), and deployment view (structure of the deployment set)

The architecture team is responsible for system-level quality, which includes attributes such as reliability, performance, and maintainability. These attributes span multiple components and represent how well the components integrate to provide an effective solution. In this regard, the architecture team decides how most multiple component design issues are resolved.

### 6.3 Software Development Team

The software development team is the most application specific group.

**Life-Cycle Focus**

| **Inception** | **Elaboration** | **Construction** | **Transition** |
|---|---|---|---|
| Prototyping support | Critical component design<br>Critical component implementation and test<br>Critical component baseline | Component design<br>Component implementation<br>Component stand-alone test<br>Component maintenance | Component maintenance<br>Component documentation |

In general, the software development team comprises several sub-teams dedicated to groups of components that require a common skill set. Typical skill sets include the following:

- Commercial component: specialists with detailed knowledge of commercial component central to a system architecture.
- Database: specialists with experience in the organization, storage and retrieval of data.
- Graphical user interfaces: specialists with experience in the display of organization, data presentation, and user interaction necessary to support human input, output and control needs.

ISSN 1512-1232

- Operating systems and networking: specialists with experience in the execution of multiple software objects on a network of hardware resources, including all typical control issues associated with initialization, synchronization, resource sharing, name space management , reconfiguration, termination and inter-object communications.
- Domain applications: specialists with experience in algorithms, application processing, or business rules specific to the system.

### 6.4 Software Assessment Team

There are two reasons for using an independent team for software assessment, The first has to do with ensuring an independent quality perspective. This often debated approach has its pros(such as ensuring that the ownership biases of development do not pollute the assessment of quality) and cons (such as relieving the software development team of ownership in quality , to some extent.) A modern process should employ use-case-oriented or capability-based testing (which may span many components) organized as sequence of builds and mechanized via two artifacts:

1. Release specification (the plan and evaluation criteria for a release)
2. Release description(the results of a release

***Life-Cycle Focus***

| **Inception** | **Elaboration** | **Construction** | **Transition** |
|---|---|---|---|
| Infrastructure planning scenario prototyping | Infrastructure baseline Architecture release testing Change Management Initial user manual | Infrastructure upgrades Release testing Change Management User manual baseline Requirements Verification | Infrastructure maintenance Release baselining Change management Deployment to users Requirements verification |

Some component tests may get elevated to evaluation criteria, with their results documented in release descriptions. Many components may undergo only informal component testing by the development team, with the results captured only within the test software built by a developer. Formal testing for many components will then be subsumed in higher level evaluation criteria and corresponding release descriptions. All components are not create equal: Some of them deserve formal component testing to verify requirements, while others are best tested in the context of capability testing.

The assessment team is responsible for the quality of baseline releases with respect to the requirements and customer expectations. The assessment team is therefore responsible for exposing any quality issues that affect the customer's expectations, whether or not these expectations are captured in the requirements.

## 7 CONCLUSION

Software development team are motivated by career growth, job satisfaction and the opportunity to make an impression and get recognition. Software development work extends itself to many large domains and hence the need for order and rule should be prevalent. The discipline of software development process and management extends itself in full to planning, automation, and project control. It is believed that the project control activities act as the "senses" of the project. They are the parameters to determine the true health of the plan and make a review of the disparity between what had been planned earlier and what had been accomplished. Remedial measures must be set in at the appropriate time and implemented at all levels. Involvement and interest by the development should be sustained throughout and they need to be properly encouraged on this count

ISSN 1512-1232

## 8 REFERENCES


1) Pressman, Scott (2005), Software Engineering: A Practitioner's Approach (Sixth, International ed.), McGraw-Hill Education.
2) David I. Cleland, Roland Gareis (2006). Global project management handbook. McGraw-Hill Professional, 2006. ISBN 0071460454. Pp.1-4.
3) Martin Stevens (2002). Project Management Pathways. Association for Project Management. APM Publishing Limited, 2002.
4) Morgen Witzel (2003). Fifty key figures in management. Routledge, 2003. ISBN 0415369770. Pp. 96-101.
5) Bjarne Kousholt (2007). Project Management –. Theory and practice.. Nyt Teknisk Forlag. ISBN 8757126038. p.59.
6) F. L. Harrison, Dennis Lock (2004). Advanced project management: a structured approach. Gower Publishing, Ltd., 2004. ISBN 0566078228. p.34.
7) Stellman, Andrew; Greene, Jennifer (2005). Applied Software Project Management. O'Reilly Media. ISBN 978-0-596-00948-9.
8) Albert Hamilton (2004). Handbook of Project Management Procedures. TTL Publishing, Ltd. ISBN 07277-3258-7.
9) Edward Kit, Software testing in the real world, Addison-Wesley publications, 2000, ed.1
10) Pankaj Jalote, An integrated approach to software engineering, Narosa publications, 1997, ed. 2
11) Shari Lawrence Peleeger, software engineering theory and practice, Pearson education, 2001, ed. 2
12) Richard Fairly, Software engineering concepts, McGraw-Hill Inc.,1985
13) Myers, Glenford J. (1979). The Art of Software Testing. John Wiley and Sons. p. 145-146. ISBN 0-471-04328-1.
14) Barry W. Boehm, Software Engineering Economics, Prentice-Hall Inc., 1981.
15) Diomidis Spinellis. Code Quality: The Open Source Perspective. Addison Wesley, Boston, MA, 2006.
16) Ho-Won Jung, Seung-Gweon Kim, and Chang-Sin Chung. Measuring software product quality: A survey of ISO/IEC 9126. IEEE Software, 21(5):10–13, September/October 2004.
17) Stephen H. Kan. Metrics and Models in Software Quality Engineering. Addison-Wesley, Boston, MA, second edition, 2002.
18) Jeff Tian, Software Quality Engineering: Testing, Quality Assurance, and Quantifiable Improvement, IEEE Computer Society Press, 2005, ISBN: 0471713457.
19) Musa, J.D, A. Iannino, and K. Okumoto, Engineering and Managing Software with Reliability Measures, McGraw-Hill, 1987
20) Dustin, Elfriede (2002). Effective software Testing. Addison Wesley. p. 3. ISBN 0-20179-429-2


**Article received: 2010-04-29**