1

Towards the 21st Century's Software, State of the art and International Standard in JTC1/SC7

Motoei Azuma
Department of Industrial Engineering and Management, Waseda University
3-4-1, Okubo, Shinjuku-ku, Tokyo 169, Japan

1. INTRODUCTION

Efforts towards better software quality have been made for over three decades. In sixties to early seventies (1960 - 1971), most of the efforts were focused on testing and debugging. Then software engineering emerged. The following latter seventies and early eighties (1972 - 1982) are considered to be the age of software engineering, especially structured techniques such as structured analysis, design, coding and testing. After then (1983 - 1994) CASE tools have been the key topics driven by such technologies as Object Oriented, Artificial Intelligence and Graphical User Interface. Unix based workstations and personal computers have made great progress during this period. Another remarkable advance have made in the field of software management including software metrics, TQM (Total Quality Management) in software and process improvement.

Nevertheless, much efforts have yet to be made. This paper focuses on recent technologies in the area of software management and related international standard in ISO/IEC JTC1/SC7.

2. APPROACH AND MODEL

There are three approaches for improving software quality and productivity. They are information science approach, industrial engineering approach and human factors approach. Software management is an industrial engineering approach which applies industrial engineering technologies into software production. The main goal of software management is to develop or maintain a better quality software product at a lower cost by a target date. However, it is not enough. Managerial effort to improve the organization, where staff are working in better environment and trying to achieve better results, is considered to be more important.

The goals of software management are metrication, standardization, and automation (Figure 1). Metrication means to make an attribute of management object to be measurable. In every area of industry, if they want to manage any object, such attributre of the objects as productivity, human performance, cost and quality must be measurable.

Standardization is the important driving force for better management. Both product standards and process standards have contributed to the progress in various industries. Product standards helped consumer to select a product from wider variety of products, and helped to aquire a parts for maintenece. Process standards made it possible that the process be automated and that inexperienced workers make a good job.

Automation means to automate effort required for management, such as data collection and analysis. By automation a manager can concentrate on decision making for exceptional matter and risk management.

The author developed a cause effect model (Figure 2) which shows relation between environmental factors, process and results of the process, and made joint questionnaire survey on software management in EC (EU) and Japan based on it. There are both external and internal environmental factors which are directly influential on the process or have indirect influence

M. Lee et al. (eds.), *Software Quality and Productivity*

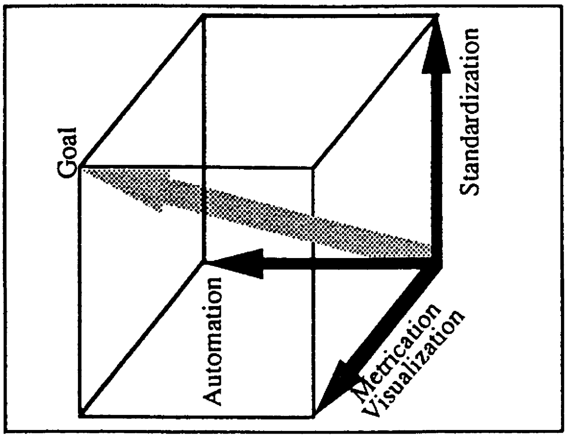

Fig.1 Management Approach

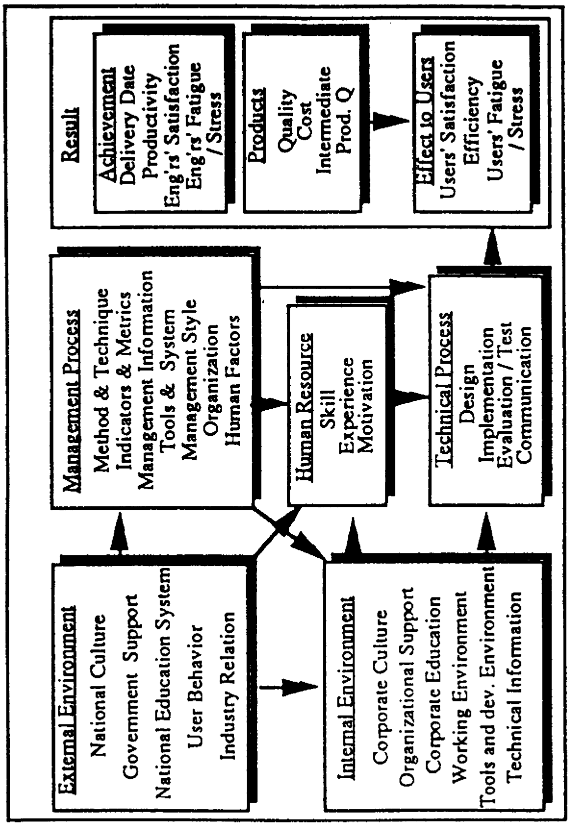

Fig.2 Software Management Cause-Effect Model

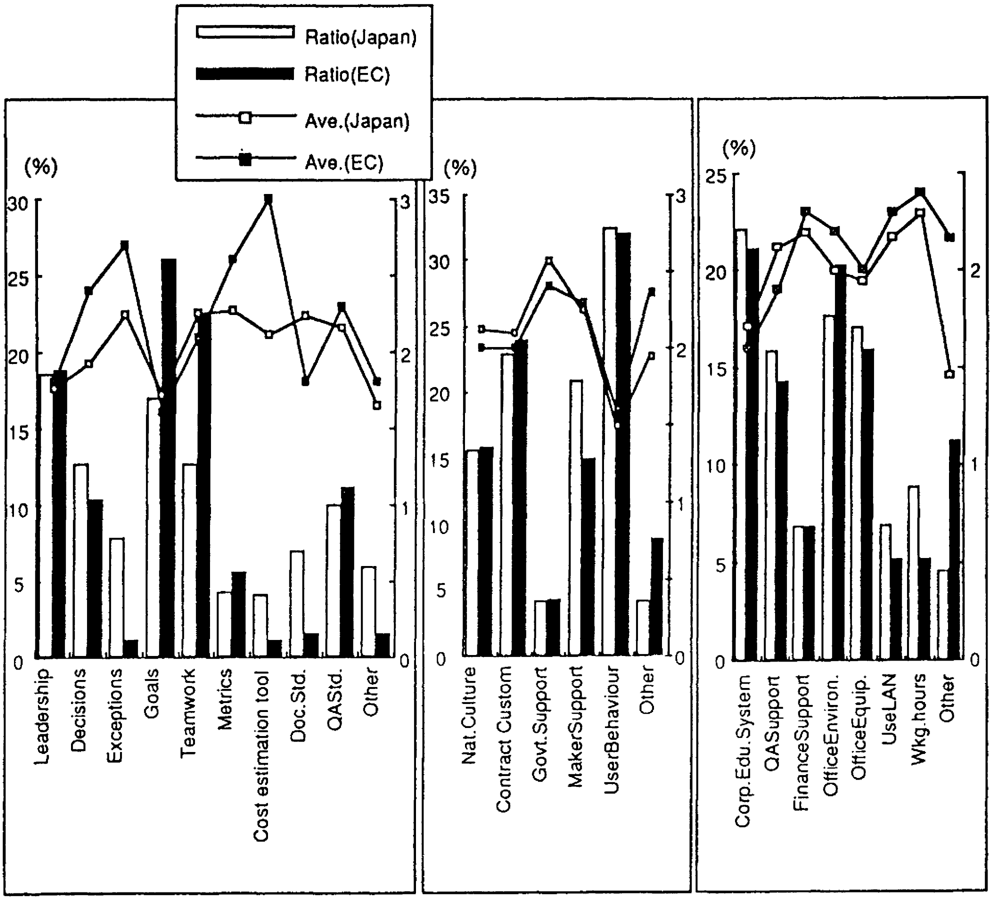

Fig.3a Influential Management Factors

Fig. 3b Influential External Factors

Fig.3c Influential Supporting Functions

through management or human factors. Some interesting results, such as influential factors were clarified (Figure 3) [1].

3. ENVIRONMENT AND SUPPORT

3.1 Organizational Support

Because of very rapid growth in information technologies, a software engineer has too many things to study and the volume of new information in the area beyond human capability to handling. Therefore the support by professional staffs is vitally important. A successful organization receives organizational support from such staffs as software engineering laboratories, fiance department and quality assurance department. Some companies have software engineering laboratories for developing and transfering technologies. Quality assurance and Software engineering are selected as important organizational supporting functions by EC/J survey. Corporate education function is also recognized as important. Most of Japanese large company have good corporate education system. External consultants may be helpful especially for those companies which are not large enough to have independent supporting organizations. Guides for these supporting functions should be established soon.

3.2 Tools Environment Support

Workstations and personal computers have made grate progress in cost performance. UNIX and MS-DOS have been also successful. GUI (Graphical User Interface) is now very popular, and Macintosh. MS-Windows and X-Window are widely used. In earlier days, UNIX based programming support environments were centralized, but they became personalized. And recently these personalized environments are connected to each other with local and wide area network and form distributed systems. This distributed environment is more difficult and time consuming to manage. Yet many organizations do not have any professional staff for system management support.

Many kinds of tools to support software works were developed. Among them, CASE (Computer Supported Software Engineering) tools are getting well-known. But, unfortunately, because of wrong pricing policy (too expensive), they are not widely used. Because of diversity of computer hardware, network and software, recent tools and their environment are not so easy to integrate and maintain. That is why tools integration and management support is also selected as important supporting function.

Standards such as CDIF (CASE Data Interchange Format) by CDIF Technical Committee in USA and PCTE (Portable Common Tool Environment) by ECMA(European Computer Manufacturers Association) TC33 in Europe are expected to solve this problem. SC7/WG11 has many projects in this field. SC7 decided to accept CDIF documents as onputs to the WG11 works. SC7/WG4 has projects on CASE tool selection and implementation.

4. PROCESS ASSESSMENT AND IMPROVEMENT

One of the essential concept of traditional Japanese TQM is to improve quality by process. This means that a person in charge of a process is also responsible for quality. He/she must think the person in charge of the next process as a customer. So he/she must try to improve the process. Major Japanese computer mainframers have made efforts to apply this concept into software. SWQC in NEC, which tried to improve a process by means of QC group activity, is well known as a successful case.

Process assessment has become suddenly fashionable since SEI published its method [2]. There are two possible reasons of it. Various attempts to analyze Genjo (current status) of software process and make Kaizen (improvement) have been made in many companies where TQM was applied in software. These companies needed more formal checklists, criteria for Genjo Analysis and guidelines for Kaizen. Other companies needed more generic guidelines for process improvement. SEI method met these requirement. However, it is still at the entrance of the technology development for process improvement. The technology to make the human process evollutional, and the environment which adapt to the process are possible goals for it. IEEE transaction on Software Engineering had a "Special section on the evolution of software process" in December 1993, and IEEE Software edited "Measurement Based Process Improvement" issue in July 1994. Many conferences and workshops on software process, such as ICSP (International Conference on Software Process), ISPW (International Software Process Workshop) and European Workshop on Process Modeling have been held.

In JTC1/SC7, WG7, WG8 and WG10 are working on developing standards in the software process field. DIS 12207 Software Life Cycle Process was developed by WG7 and circulated to the P-member bodies for the final ballot closing on August 28. WG8 has projects on configuration management and quality assurance. WG10 is working on developing a series of standards on process assessment.

5. PRODUCT QUALITY EVALUATION AND METRICS

Many works have been done in this field individually or sometimes relatedly. They include quality model, Metrics and methodologies and their supporting tools. The followings are some examples of them.

5.1 Quality Model

The concept and technique of quality deployment is well known as one of the important techniques in Japanese TQM. When we evaluate quality of anything, for several characteristics. For example, quality of a car is evaluated taking such characteristics as, e.g., reliability, performance, safety, comfort and drivability, into consideration. Relative weight (Importance) of each characteristics varies depending on nature of the target and user's purpose. Quality characteristics (or factors) can be deployed. This structure is called quality model.

Boehm [3], McCall [4], developed quality model in 1976 and 1977 respectively. Several models followed in Japan. Based on these experiences, INSTAC (Information Technology Research and Standardization Center) of JSA (Japanese Standards Association) made a five-year-research on quality evaluation standardization. The model developed by INSTAC was used as a input for ISO/IEC 9126. ISO/IEC 9126 (Information technology - Evaluation of software product - Quality Characteristics) defines six quality characteristics and subcharacteristics [5]. ISO/IEC 9126 has been widely used even before it was published as international standard.

5.2 Metrics and Indicators

There are only a few well known metrics that are used. Examples are Complexty metrics, Cyclomatic number, module metrics, such as, Fan-in and Fan-out, Cohesion and Coupling [6], [7], [8]. However, only a limited number of companies are using these metrics.

In order to utilize these metrics, it is important that there are some guides to utilize these metrics for specific purpose and guides to develop a metric when necessary. And engineers must be well educated. ESPRIT (European Strategic Program for Research in Information Technology) supported a project METKIT (Metrics Education and Training KIT) which developed a tool kit for metrics education [9].

INSTAC developed a series of indicators and metrics which were contributed as an initial input to support JTC1/SC7 project 7-13 (ISO/IEC 9126 Series).

Function Point is another topics for standardization in the area of software metrics. Though FP is not a metrics to measure quality but a metric to measure a size of software, it is important not only for scale and cost estimation of a project but also for quality management, because many quality indicators are normalized by a scale of a software. SC7/WG12 is a very new working group to standardize the Function Point supported by International Function Point User Group.

5.3 Quality Requirement

In order to develop or acquire better quality software, requirement should be stated clearly for every quality characteristic. It is desirable that requirement is shown with quantitative evaluation criteria using metrics. However, most of requirement technologies are focusing only on functional requirement, and sometimes on efficiency requirement. Usability is an another area of topics but not in software engineering community but in ergonomics community. There are several good check lists developed by Shneiderman and ESPRIT which are useful for both usability requirement and review [10], [11].

Six quality characteristics and subcharacteristics defined in ISO/IEC 9126 can be applicable for this purpose. ISO/IEC 12119 (Information technology - Quality Requirement and Testing) describes quality requirement and testing directives based on ISO/IEC 9126, which are also applicable if the target software is a consumer software. However more effort is necessary to develop and validate indicators and metrics to state quality requirement and criteria qualitatively.

5.4 Evaluation Methodology

In USA, Murine [12] developed SQM (Software Quality Metrics) based on McCall model. He placed a Metric level under the criteria of McCall Model and developed metrics, though most of the metrics are subjective and ad hoc. A metric is a rule and method to measure an attribute of software. Basili et al. are organized the TAME project [13] and are developing a system for quality management. Their approach is called GQM, that is, Goal, Question and Metric. Nance and Arthor of Virginia Polytechnic developed a methodology sponsored by US Navy [14].

In Japan Azuma (the author) et al developed SQMAT [15], and IBM Japan developed SQULAS in 1982. based on Japanese traditional quality deployment approach. Since then, major Japanese computer companies have developed its own quality system, which caused new problem to the software users. Miyoshi et al. [16] tried to evaluate software development environment quality.

In Europe, ESPRIT project took major role in developing information technologies. ESPRIT had another project on software metrics called SCOPE. SCOPE was used as a initial input to the Evaluators' Guide written bellow.

JTC1/SC7/WG6 (Convenor: M. Azuma) is developing the following series of guides for evaluation of software product to support ISO/IEC 9126.

Part 1	General Guide (Co-editor: M. Azuma, N. Bevan, S. Nishiyama)
Part 2	Managers' Guide (Editor: T. Ipoly)
Part 3	Developers' Guide (Editor: H. Schafer, J. Boegh)
Part 4	Buyers' Guide (Editor: J. Harauz)
Part 5	Evaluators' Guide (Editor: P. Robert)
Part 6	Evaluation Module Guide (Editor: J. Boegh)

6. HUMAN FACTORS

Human capability is more diverse in mental area than physical are. An Olympic marathon runner can run two times faster than a city runner in speed. However, expert programmer can be 30 times efficient than novice programmer in productivity. Therefore more effort should be focused on this area including motivation, team, human relation, education and training, and working environment.

Software psychology is one of the most important research area in human factors where B.Shneiderman made initial contribution [17]. Dr. Fujigaki made interesting research in the Techno-stress field [18] and published a book (in Japanese) on it. Concerning the team structure, once the chief programmer team, egoless team and matrix team and so on were area of interests. But it seems that most of researcher in the area lost their interest in this field.

Working environment was another topics. The author proposed the initial idea of Satellite Office. The experimental office was built in a suburb of Tokyo [19]. It aimed to offer software engineers such environment that one or a small group of software engineers can work in a satellite office which is located in urban residential area, outside of the main office, co-operatively with engineers in the main office or in another satellite office as if they were working by sitting side by side in the same office.

A CSCW (Computer Supported Co-operative Work) and a groupware are geting growing interest and concern of many researchers, manageres and users [20], [21]. But it seems that definitions of these terms have still not concensus. CSCW is a generic name of the concept or work style which represents that a group of people work co-operatively by means of computer support. Groupware is a name for the technology or software to support CSCW. CSCW/ Groupware is one promising technology area for supporting software development and maintenance process, becuse still software process is man power intensive work and it should be co-operative. CASE tools must have CSCW feature in near future. However, it will take longer before a new work item for international standard in this area will be proposed.

7. CONCLUSION

Conputer science and engineering approaches, industrial engineering and management approaches and human factors approaches are not exclusive but complementary or co-operative to each other. Advaces were made in these area, such as CASE tools, software metrics, quality assurance and CSCW for the past ten years. Some of them were co-operative. It is important that new technologies are harmonized and that researchers in this field will always keep it in their mind. Efforts for standardization will be increased and the process will be accelrated.

However, because of reasons as follows, more effort must be done for the quality and productivity improvement of software.

(1) Emerging new technologies in information systems, such as multi media and vertual reality, require new software engineering. New concepts may be necessary for this problem. There is egg and chicken relation between information technologies and software engineering.

(2) No matter how deliverately requirement is analyzed, it will change. Evolutional software process paradigm and its supporting environment and adaptive software look like promising solutions.

(3) Human brain process is difficult to understand. Developing a CASE tool with groupware functionality and adaptive to the user looks like promising alternative solutions.

REFERENCES

1 Software Management Practice and Metrics EC and Japan, Proc. AQuIS 93

2 W.S.Humphrey, Managing the Software Process, Addison-Wesley, 1989

3 B.W.Boehm et al., Quantitative Evaluation of Software Quality, Proc. 2nd ICSE(International Conference on Software Engineering), pp592 - 605

4 McCall et al., Rome Air Develop Center Report, TR-77-369, 1977

5 ISO/IEC 9126: Information technology - Evaluation of software - Quality characteristics and guides for their use.

6 R.Dumke, Softwareentwicklung nach Masz, Vieweg, 1992

7 N.E.Fenton, Software Metrics A rigorous approach, Chapman and Hill, 1991

8 K.H.Moller, and D. J. Paulish, Software Metrics, Chapman and Hill, 1993

9 M.Bush and M.Russel, Software Engineering Measurement: A Modular Course, Proc. COMPSAC 91, pp571 - 576

10 Schneiderman, Designing the user interface, second Edition, Addison-Wesley, 1992

11 S.Ravden and G.Johnson, Evaluating Usability of Human-Computer Interfaces, Ellis Horwood,1989

12 G.E.Murine et al., Measuring Software Product Quality, Quality Progress, Vol.17, No.5, pp16-20, 1984

13 V.R.Basili and H.D.Rombach, The TAME Project : Towards Improvement-Oriented Software Environments, IEEE Trans. on Software Engineering, Vol.14 No.6, pp758-773, 1988

14 Arthur, J.D. and Nance, R.E., Developing an automated procedure for evaluating software development methodologies and associated products- A final report, Technical report SRC-87-007, Systems Research Center and Virginia Tech, 1987

15 T.Sunazuka and M.Azuma, Software Quality Measurement and Assessment Technology, Proc. 8th ICSE, pp142 - 148, 1985

16 Miyoshi, Azuma,An Empirical Study of Evaluating Software Development Environment Quality, IEEE Tr. SE Vol. No.

17 B.Shneiderman, Software Psychology, Winthrop, 1980

18 Y.Fujigaki, Interactive Approach to Reduce the Stress of Working Group in Information Technology, Proc. 11th Congress of International Ergonomics Association, pp1265-1267, 1991

19 K.Fujino et al., The Near Future Office Environment in the C&C Era - C&C Satellite office -, NEC RESEARCH & DEVELOPMENT, No.81, pp1-7, 1986

20 D.Diaper and C.Sanger Eds., CSCW in Practice: An Introduction and Case Studies, Springer-Verlag, 1993

21 I.Greif Ed., Computer-Supported Cooperative Work: A Book of Readings