Total quality management in software development process

Eldon Li
National Chengchi University
96 PUBLICATIONS **1,701** CITATIONS

Houn-Gee Chen
National Taiwan University
57 PUBLICATIONS **1,209** CITATIONS

The Journal of Quality Assurance Institute, Vol. 14, No. 1, January 2000, pp. 4-6 & 35-41.

Total Quality Management in Software Development Process

Eldon Y. Li, *California Polytechnic State University, USA*

Houn-Gee Chen, *National Chung Cheng University, Taiwan*

Waiman Cheung, *The Chinese University of Hong Kong, PRC*

Abstract

This paper discusses the essences of total quality management (TQM) concept and identifies the principles of successful TQM implementation. It contrasts the Quality Seven (Q7) and the Management Seven (M7) tools commonly used in the TQM process. It also describes the Deming's quality management concept and his fourteen-point management method. It briefly explains the similarities between software development process and product development process. Finally, it discusses how to instill Deming's TQM method in software development process and provides recommendations to TQM prospects or participants for avoiding pitfalls and ensuring success during TQM implementation.

Keywords: Total quality management, continuous process improvement, statistical process control, software development processes, Deming's management methods.

1. Introduction

The total quality management (TQM) concept represents a fundamental change in the definition and treatment of quality in product development. Traditionally since the beginning of the industrial revolution, US industries had a product-focused mentality. The philosophy of "if I make it, someone will buy it" was prevalent among U.S. manufacturing companies. A noticeable follower of this philosophy was the U.S. auto industry during the oil crisis back in the late 1970's. Many US auto manufacturers were hit very hard because they continued to manufacture large inefficient cars that people could not afford to drive it. Smaller, more efficient foreign cars flooded the market, cutting in on US automakers who believed Americans would continue to buy American cars despite the rapidly rising gasoline price. Their unwillingness to satisfy customer needs had cost them billions of dollars. Under this traditional philosophy, the view of quality is as follows [Strickland, 1988]:

1) **Productivity and quality are conflicting goals.** Improving quality consumes additional corporate resources that are needed to maintain productivity. Therefore, quality can be improved only at the expense of productivity.
2) **Quality is defined as conformance to specifications or standards.** Such conformance pays no attention to incorrect specifications or obsolete standards that are prevailed in most companies.
3) **Quality is measured by degree of non-conformance.** It is usually measured by the defect count in "parts per million"— the famous six-sigma measurement. Such measurement focuses on the degree of non-conformance in stead of customer satisfaction.
4) **Quality is achieved through intense product inspection.** Such inspection consumes much of the corporate resources. If a product fails the inspection, it needs to be reworked or scraped.
5) **Some defects are allowed if a product meets minimum quality standards.** This implies that customers are willing to pay for a "buggy" yet working product.
6) **Quality is a separate function and focused on evaluating production.** It is assumed that the production group will welcome such independent quality function.
7) **Workers are blamed for poor quality.** However, replacing a worker does not mean improving quality. Furthermore, poor quality may come from the supplier side.
8) **Supplier relationships are short-term and cost-oriented.** There is no way to control the quality of raw materials or parts delivered by the suppliers.

This view of quality has taken a dramatic turn since the emergence of TQM concept in the early 1980's. In fact, TQM is by no means an overnight invention. It is a combined teachings of various quality experts. The concept germinated in the 1920's when Walter Shewhart [1931, 1939] of Bell Labs introduced statistical controls to combat and reduce variability in testing and experimentation. Throughout the postwar 50's, W. Edwards Deming [1981, 1982, 1986] took his knowledge into the Far East and established blueprints of organization-wide quality control for fledgling Japanese businesses.

Joseph Juran [1945, 1951] followed a similar route into Japan by introducing total quality control in production. Kaoru Ishikawa [1976, 1985] took quality a step further in the 1960's and advocated that quality should be the responsibility of individual employee and that top management should provide quality leadership. He introduced the concept of Quality Control Circles. Genichi Taguchi [1986, 1987] focused on the use of quantitative methods and design of experiments to improve quality, particularly in the area of product design. On the U.S. side, Armand Feigenbaum [1951, 1957, 1961] introduced the concept of total quality control in the late 1950's. In the late 1970's, Philip Crosby [1967, 1969, 1979] began pushing zero defects as the default standard of performance, and not the exception. Under the TQM culture, the traditional goals of maximizing profit or benefit, minimizing costs, and achieving controlled growth have been replaced by improving customer satisfaction, improving quality, and reducing schedule. The purpose of this article is to review the TQM concept, identify the principles of TQM implementation, introduce Deming's management method, and apply the method to software development processes.

2. Total Quality Management

The Definition

According to the *Webster's Dictionary,* "quality" is "a degree of excellence; a distinguishing attribute." That is, quality is the degree to which a product lives up to its performance, endurance, maintainability, and other attributes that a customer expects to receive from purchasing this product. In order to produce quality product, one must instill TQM concept into one's product development process.

The word "total" means the total of everything in an organization. That is, it covers every process, every job, every resource, every output, every person, every time and every place. According to the American Society for Quality Control (ASQC), total quality management (TQM) "is a management approach to long-term success through customer satisfaction. TQM is based on the participation of all members of an organization to improving processes, products, services, and the culture they work in. TQM benefits all organization members and society. The methods for implementing this approach are found in the teachings of such quality leaders as Philip B. Crosby, W. Edwards Deming, Armand V. Feigenbaum, Kaoru Ishikawa, and J.M. Juran." [Bemowski, 1992] The Department of Defense also provides a concise and accurate definition as below:

"TQM is both a philosophy and a set of guiding principles that represent the foundation for a continuously improving organization. TQM is the application of quantitative methods and human resources to improve the material and services supplied to an organization, and the degree to which the needs of the customers are met, now and in the future. TQM integrates fundamental management techniques, existing improvement efforts, and technical tools under a disciplined approach focused on continuous improvement." [Department of Defense, 1991]

TQM View of Quality

Under the TQM concept, quality is defined and judged by the customer. Therefore, it acknowledges a customer-driven economy. It focuses on continuous process improvement to achieve high quality of product (or service). Its strategy tries to achieve "total quality" throughout the entire business, not just in the product. It suggests that any improvement that is made in the business, be it a better design of a component or a better process of a system, will help to improve the "total quality" of the organization and the quality of the final product. Under this philosophy, the view of quality is very different from the traditional one:

1) **Productivity and quality are not conflicting goals.** Productivity gains can be achieved through quality improvements. Better quality of product and process will reduce rework, errors, and waste. This, in turn, improves the productivity.
2) **Quality is correctly defined requirements that satisfy users needs.** The ultimate quality of a product is its ability to satisfy user's needs. One should take one step further to get the consumer involved in defining the product requirements. It is plausible to say that quality is defined and judged by the customer.
3) **Quality is measured by user satisfaction as well as by continuous process and product improvement.** Just as one would expect, customers prefer to purchase software that fits their needs and performs beyond the quality standards. The TQM practice shuns the old adage of "don't fix it if it ain't broke."
4) **Quality is achieved by effective product design and process controls.** Relying on product inspection implies that errors will definitely be made. Quality cannot be achieved by inspection. It should be built in, not added on. To build in quality, one must perform effective product design and process controls.
5) **Defects are prevented through process control techniques.** Zero defect and perfection of processes should be the goals if a company wishes to keep improving quality.

6) **Quality is a part of every function in all phases of the product life cycle.** It simply does not make sense to go about production haphazardly or without a quality-laden plan and expect a quality good or service and a happy customer for that matter.
7) **Management is responsible for quality.** Only management has the authority to change the working conditions and processes, and only management has the knowledge to coordinate quality function in all phases of the product life cycle. Therefore, management should be responsible for quality, not the workers.
8) **Supplier relationships are long-term and quality-oriented.** Suppliers are just as an important part of the team as any other members. Since management is responsible for quality, it must also take charge of building long-term and quality-oriented relationships with suppliers.

3. Principles of TQM Implementation

Based on the lessons learned by various companies implementing TQM as reported in the literature, there are some principles of practices that are instrumental to the success of TQM implementation [Department of Defense, 1990; Moore, 1990]:

1) **Quality is everyone's business.** Quality is the concern of not only the management but also the workers. By empowerment, that is empowering employees with the ability to stop production if quality is sacrificed, quality can be dramatically improved. Workers feel a sense of belonging to the process and a pride in the quality of their work. Quality is perceived as a team effort. Auto factories are now empowering their employees with the ability to stop the whole production line if someone discovers a quality defect.
2) **Customer Emphasis.** One must focus on satisfying internal and external needs, requirements, and expectations, not just on meeting specifications. This is essentially creating a customer focus. Since customers are the ones that drive production, their needs and expectations should be the focus of all improvement efforts. Customers are not only those who buy finished products. There are also workers within the company who use the components produced by other workers. These workers are internal customers. A production line can be conceived as having a string of customers, starting with the one who is making the contract. Each person is responsible for improving the quality of the product that they pass on to the next customer. Under the TQM culture, everyone has a customer.
3) **Quality must be built into the product.** Quality cannot be an afterthought. It must be constantly measured and quantified. The question: "Is this good quality?" must be a centerpiece in any development project. It must be a concern from beginning to end. Quality is not determined by picking the best of the bunch after production and recycling the bad ones. "Bad ones" should never exist in a TQM environment. Defects should be discovered before any production occurs. This is accomplished by building quality into the product. It is easiest to understand this concept by thinking about quality as a part of a product. The product cannot work without the quality component installed. Therefore, during every stage of the development process, the developer must ask himself: "Have I installed the quality component in this product?"
4) **TQM requires management commitment and involvement at all levels**. TQM must be implemented from the top down in every organization. If management does not have a commitment to a TQM culture, it will fail. The management must provide leadership in implementing the change; the workers do not have the power to do so. Do not blame the workers for poor quality; the management and the systems are responsible for quality.
5) **TQM accomplishment involves continual training.** Continuous improvement includes the improvement of one's ability in performing one's job. An employee must be trained in TQM principles and in the tools and techniques for implementing TQM. Such training credential should be treated as an accomplishment for performance evaluation.
6) **Leadership is substituted for slogans and exhortations.** We have all heard the slogan: "Quality is job one." This example is from Ford Motor Company. However, slogan means nothing when we say it but we do not do it. Ford still produces buggy cars despite its great success in *Team Taurus* project [Walton, 1986]. What Ford needs is better leadership in quality improvement.
7) **Long-term emphasis on measurable processes and productivity improvement.** TQM cannot be implemented overnight. It is a long-term process that takes years to implement. It is a complete cultural change in the organization to focus on continuous improvement. The problem with achieving continuous improvement is that it requires measures to be compared against. Therefore, a

key element of the TQM culture is qualified metrics–measurements taken continuously in order to chart progress.

8) **Understand the current process before improvement begins.** We must understand how things work in the organization to be able to improve it. Understanding how it works involves being able to measure the process in order to compare "improvements" against it.

9) **Cross-functional orientation and teamwork.** The essence of cross-functional teams is to integrate many different parts of the organization into the development process. For instance, programmers must involve users from finance, accounting, marketing, and other departments in the development of a software product. This philosophy is developed with the thought that developing a product is not just the designer's concern. Everyone who is involved with the development, distribution, and maintenance of a product should have a say in the development of the product.

10) **Effective use of statistical methods and quality control tools.** Statistical quality control and process control techniques should be used to identify special causes of variation that are

Table 1: Quality Control Tools

Quality Seven (Q7) Tools	Management Seven (M7) Tools
1. **Cause-and-effect diagram.** This diagram is also called "fishbone diagram" or "Ishikawa diagram." It identifies, explores, and displays all possible causes or contributing factors of a problem or event.	1. **Affinity diagram.** This is essentially a brain storming output. It is based on group work in which every participant writes down his ideas and the ideas are then grouped and realigned by subject matter.
2. **Checksheet.** This document is designed to tabulate the results through routine checking of the situations. It is passed between major checkpoints during the production process and acts as a safeguard from defects.	2. **Arrow diagram.** This diagram is also called "flow chart." It is often used in PERT (Program Evaluation and Review Techniques) and CPM (Critical Path Method). It uses a network representation to show the steps necessary to implement a plan or to complete a process.
3. **Control chart.** This chart serves to detect special causes of variation. The chart has control limit lines at the center, top, and bottom levels. Sample data are plotted in dots on the chart to evaluate process situations and trends.	3. **Matrix diagram.** This diagram is used to clarify the relations between two different factors. It is often used in deploying quality requirements into counterpart (engineering) characteristics and then into production requirements.
4. **Histogram.** This diagram graphically displays a set of frequency data in bar graphs and enables evaluators to determine problems by checking the dispersion shapes, center values, and the nature of dispersion.	4. **Matrix data analysis diagram.** This diagram is used when the matrix diagram does not provide sufficiently detailed information. It is the only M7 tool that is based on data analysis and gives numerical results.
5. **Graphs.** There are many types of graphs that are useful to evaluators. Line graphs, also called run charts, are used to illustrate variations over a period of time. Bar graphs compare categorical values via parallel bars. Circle graphs, or pie charts, indicate the categorical breakdown of values relative to the total value. Radar charts assist in analyzing previously evaluated items each may have its own axis of measurement.	5. **Process Decision Program Chart (PDPC).** This diagram is commonly used in operations research community. It is a hierarchical chart that displays how an optimal alternative is arrived. It is similar to a decision tree diagram.
6. **Pareto chart.** This chart classifies problems according to cause and phenomenon. It makes use of bar graphs sorted in a prescribed order to display the relative importance of problems by selected categories.	6. **Relations diagram.** This diagram is also known as causal model diagram. It clarifies the interrelations in a complex situation involving many interrelated factors and serves to clarify the cause-and-effect relationships among factors.
7. **Scatter diagram.** This diagram is also known as X-Y chart. It displays what happens to one variable when another variable changes in order to test a theory or make forecasts.	7. **Tree diagram.** This diagram is similar to a functional decomposition chart. It is applied to show the interrelations among goals and measures and among processes and activities.

Sources: Adapted from Ishikawa [1976], Imai [1986], Brassard [1989], and Walton [1986].

points outside the control limits. Actions should be taken to remove these special causes. Moreover, any abrupt shifts or distinct trends within limits are also signals for investigation. Quality control tools such as the Quality Seven (Q7) tools and the Management Seven (M7) tools [Brassard, 1989; Imai, 1986; Ishikawa, 1976] may be used to plan for actions, collect valuable data, and chart for progress. Table 1 lists the names of these tools and their descriptions while Figures 1 and 2 display their formats. The Q7 tools are used to analyze historical data for solving a particular problem. Most problems occurring in production-related areas fall into this category. One the other hand, not all data needed for decision making are readily available and many problems call for collaborative decision among different functional areas. Under these situations, the M7 tools (also called the New Seven tools [Imai, 1986] are useful in areas such as product quality improvement, cost reduction, new-product development, and policy deployment, etc.

11) **Constant process, product, and service improvement.** A culture of constant improvement must be developed for TQM to succeed. All employees should be empowered with the ability to influence an organizational process that helps to improve quality. Once given this authority, employees must show their desire and commitment to constantly improve the company. They must be always looking for ways to improve not only their part of the organization, but also the organization as a whole. Management must foster this culture through proper reward and recognition.

12) **Incentivize TQM involvement.** Incentive is a form of position reinforcement that is the fuel of the TQM torch. Most TQM implementers use a suggestion program to solicit cost reduction ideas from employees. The ideas are evaluated by a cross-functional suggestion evaluation team and the ones with significant contributions are implemented, and the suggesters are recognized and rewarded with money and fame. Texas Instruments has a recognition function called the Quality Hall of Fame ceremonies.

13) **Information sharing.** Teamwork is the key to the success of TQM, yet it relies on sharing the necessary information and know-how among the team members and across functional areas. It has been proven that sharing such information as profit, budget, schedule, progress, errors, etc. can provide the employees a sense of ownership and importance. It encourages the employees to push themselves to work harder in order to achieve the company goals as well as their personal goals. Nonetheless, any unnecessary or problematic information such as pay scale or bonus level should not be shared because it is dysfunctional and counter-productive.

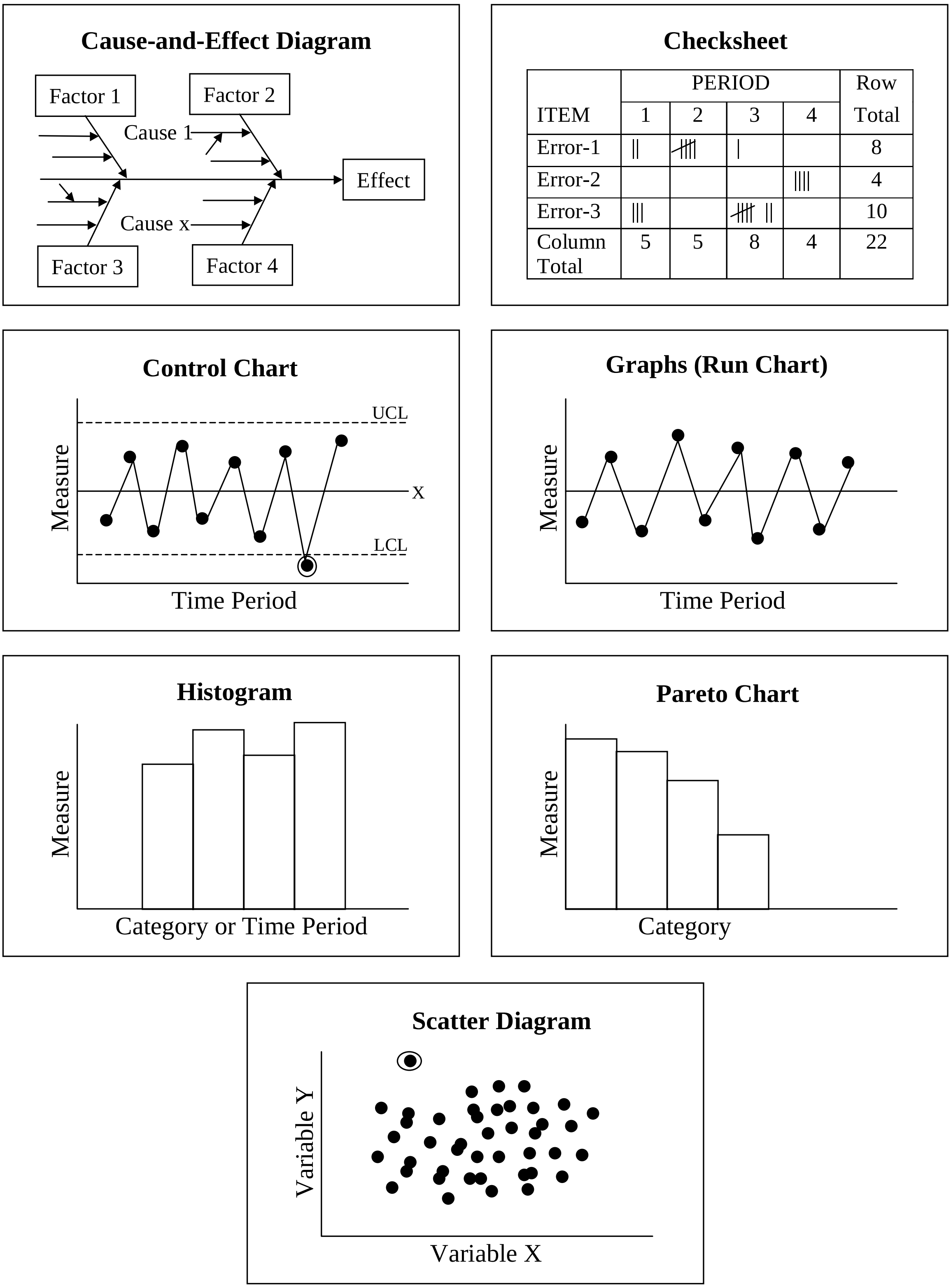
14) **Eliminate communication barriers.** Under TQM culture, there should be no communication barriers between workers and management, and between functional areas. The management must make themselves available to and easily accessible by the workers. Employee suggestion program could be implemented in order to eliminate communication barriers.

15) **Suppliers must have a TQM philosophy.** A company cannot produce a quality product if the components of which it is made are faulty. Therefore, the supplier of a company must be trained and certified as a TQM supplier. Without such a certification, any components that are purchased from the supplier cannot be guaranteed to have the quality necessary for a company to establish a TQM culture. Similar to the JIT philosophy, the TQM philosophy advocates a strong relationship with its suppliers. One should cut down the number of suppliers and provide only a few TQM certified suppliers with long-term business commitments. This motivates the suppliers to make changes for continual quality improvement and ensures that the quality of the company's products will not be sacrificed.

4. The Deming Management Method

Although Walter Shewhart is considered as the founding father of statistical quality control system, W. Edwards Deming is the first one who introduced the TQM concept. Deming offered the management his fourteen points of management obligations and identified seven deadly diseases and some obstacles of TQM implementation. The fourteen points as listed below are also known as the Deming management method [Walton, 1986].

1) Create constancy of purpose for improvement of product and service.
2) Adopt the new philosophy of total quality.
3) Cease dependence on mass inspection to achieve quality.
4) End the practice of awarding business based on price tag alone.
5) Improve constantly and forever the system of production and service.
6) Institute training on the job.
7) Institute leadership.

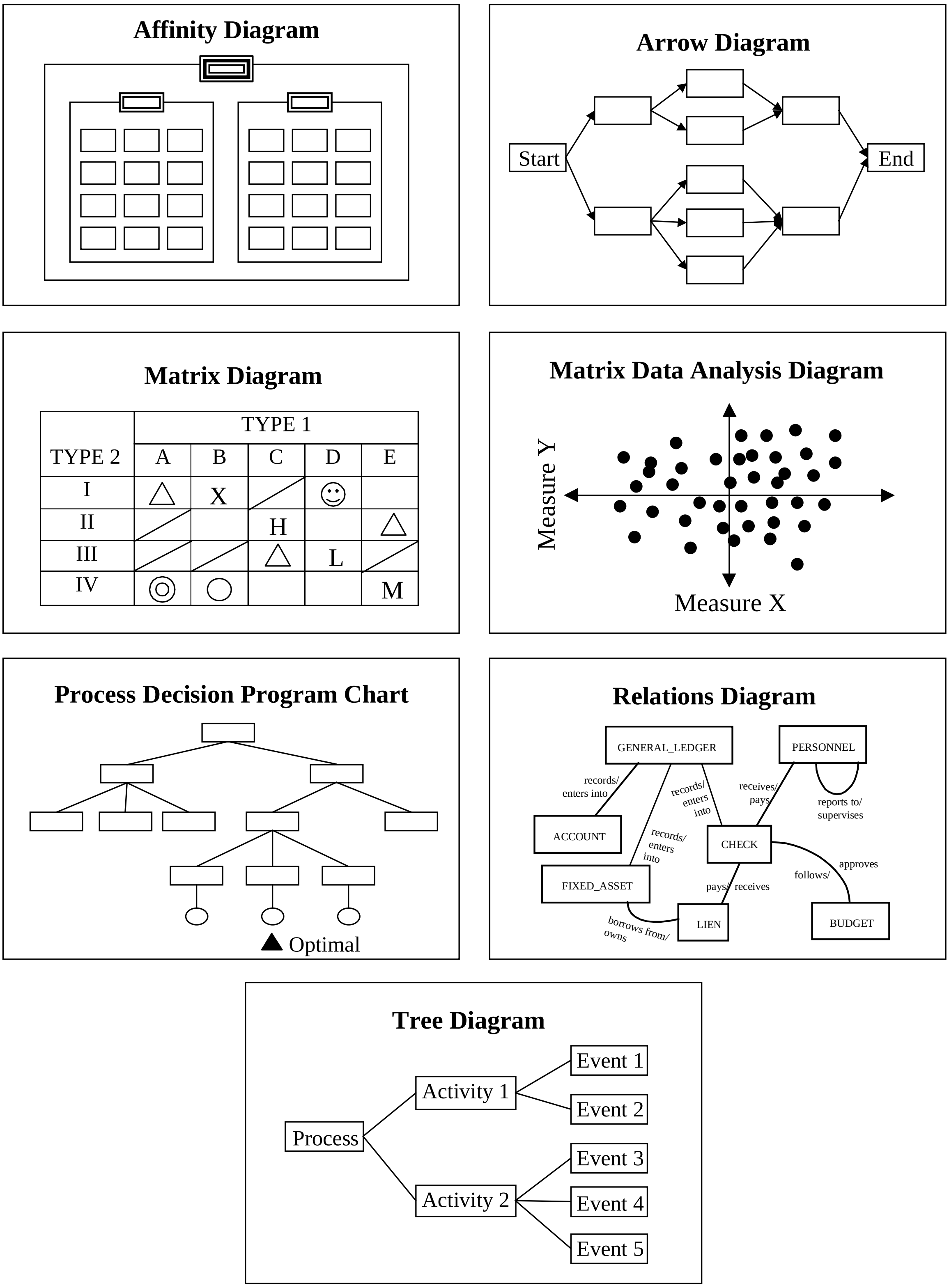


ITEM	PERIOD 1	2	3	4	Row Total
Error-1	𝍷𝍷	𝍸	𝍷		8
Error-2				𝍷𝍷𝍷𝍷	4
Error-3	𝍷𝍷𝍷		𝍸 𝍷𝍷		10
Column Total	5	5	8	4	22

Sources: Adapted from Ishikawa [1976] and Brassard [1989].

Figure 1. The Quality Seven (Q7) Tools

8) Drive out fear of job insecurity.
9) Break down barriers between departments or staff areas.
10) Eliminate slogans, exhortations, and targets for the workforce.
11) Eliminate numerical quotas, goals, and work standards.
12) Remove barriers to pride of workmanship.



Affinity Diagram

Arrow Diagram

Matrix Diagram

TYPE 2	TYPE 1				
	A	B	C	D	E
I	△	X	/	☺	
II	/		H		△
III	/	/	△	L	/
IV	◎	○			M

Matrix Data Analysis Diagram

Process Decision Program Chart

Relations Diagram

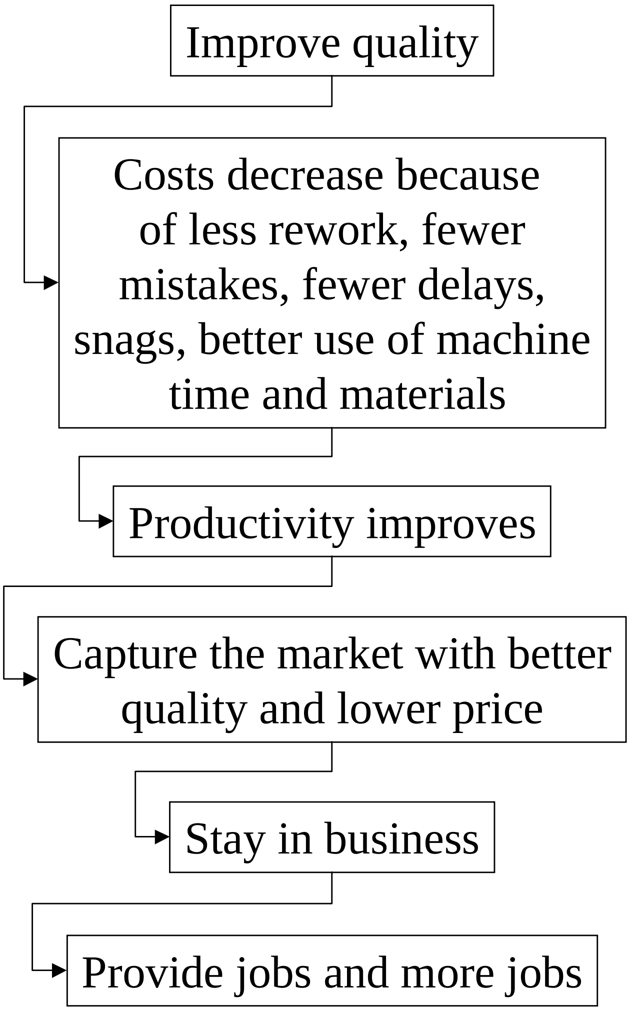
Tree Diagram

Sources: Adapted from Brassard [1989] and Walton [1986].

Figure 2. The Management Seven (M7) Tools

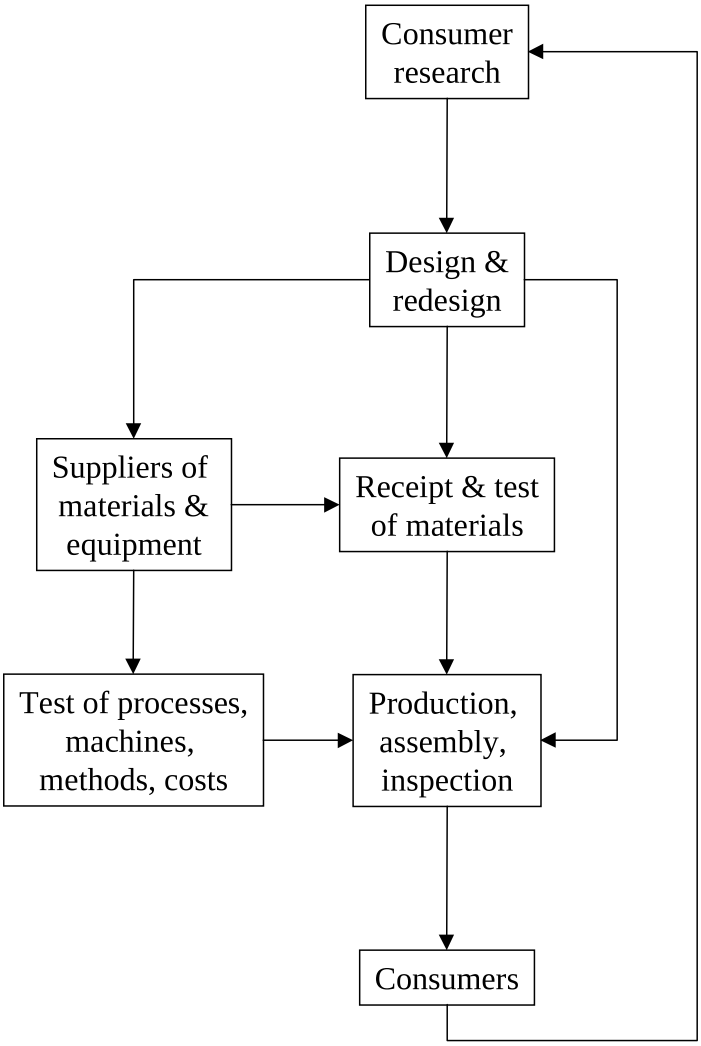
13) Institute a vigorous program of education and retraining for everyone.
14) Put everyone to work to accomplish the transformation.

Behind these fourteen points lies the philosophy of Deming's management method. This philosophy can be epitomized by the following two diagrams: the chain reaction diagram and the never-ending flow diagram [Deming, 1986]. In the former diagram (see Figure 3), Deming argues that improving quality benefits not only the company itself but also the society as a whole. It reduces costs within the company and provides more jobs to the society. In other words, instead of having everyone competing for the same piece of pie and trying to steal from one another for a bigger piece of pie, improving quality can actually make the pie bigger for everyone to enjoy. Furthermore in the latter diagram (see Figure 4), Deming emphasizes that the quality is in the eyes of the customer. Quality is to provide what the customer wants and needs. One must know through consumer research what the customer is going to use the product for. In order to meet the customer's needs, one must continually improve one's products and processes and demand the suppliers to do the same. In Deming's mind, innovation is better than revolution; quality is achieved through continually improving the organization not suddenly turning the organization


Improve quality
Costs decrease because of less rework, fewer mistakes, fewer delays, snags, better use of machine time and materials
Productivity improves
Capture the market with better quality and lower price
Stay in business
Provide jobs and more jobs

Source: Adapted from Walton [1986].

Figure 3. The Deming's Chain Reaction Diagram


Consumer research
Design & redesign
Suppliers of materials & equipment
Receipt & test of materials
Test of processes, machines, methods, costs
Production, assembly, inspection
Consumers

Source: Adapted from Walton [1986].

Figure 4. The Deming's Never-Ending Flow Diagram

upside down. In addition, total quality depends on teamwork between the company and its suppliers. If you cannot work with one supplier, do not start with another. In the long run, one should move toward one supplier for any one item so far as possible.

5. Total Quality Management for Software Process Improvement

The TQM philosophy described above can be applied to any development process, be it product development or software development.

The Product Development Life Cycle (PDLC)

Product development life cycle is a systematic and orderly approach to managing product-development activities. It usually follows the problem-solving steps prescribed by Herbert A. Simon: intelligence, design, choice, and review [Simon, 1977]. The development of a new product begins with the stage of requirements analysis. During this stage, the needs of customers are collected, analyzed, and evaluated in order to develop product specifications. Based on the customers' needs and the product specifications, design blueprints of the product are developed during the design stage. These blueprints include manufacturing design specifications and bill of materials. According to these blueprints, prototypes of the product are built and tested to evaluate the quality of the prototypes. If a prototype fails the test, the cause of failure is analyzed and identified. It might be due to flaws in prototype building process or product design process. Worst of all, it might be due to unrealistic product specifications coming from unduly feasibility studies during the requirements-analysis stage. In this case, the project of developing this new product might have to be canceled. The dashed lines in Figure 5 exhibit the cause of failure is sequentially fed back to the stage where the faulty process is. Once the prototypes passed all the tests, the best one is selected for either a pilot release (a limited-scale release to test the market) or a full release. Either release needs to develop a production line for the intended product. If the pilot release is not satisfactory, the sales information is fed back to the requirements-analysis stage and the product demand is re-evaluated. On the contrary, if the pilot release is successful, it is turned into full release and mass production of the product is performed. A stage of follow-up is proceeded. If the follow-up report indicates successful sales and profit of the product, a go-ahead signal is sent back to the full-release stage for continuing the mass production. If the report indicates otherwise, the requirements-analysis process is triggered once again and the entire product development life cycle is repeated.

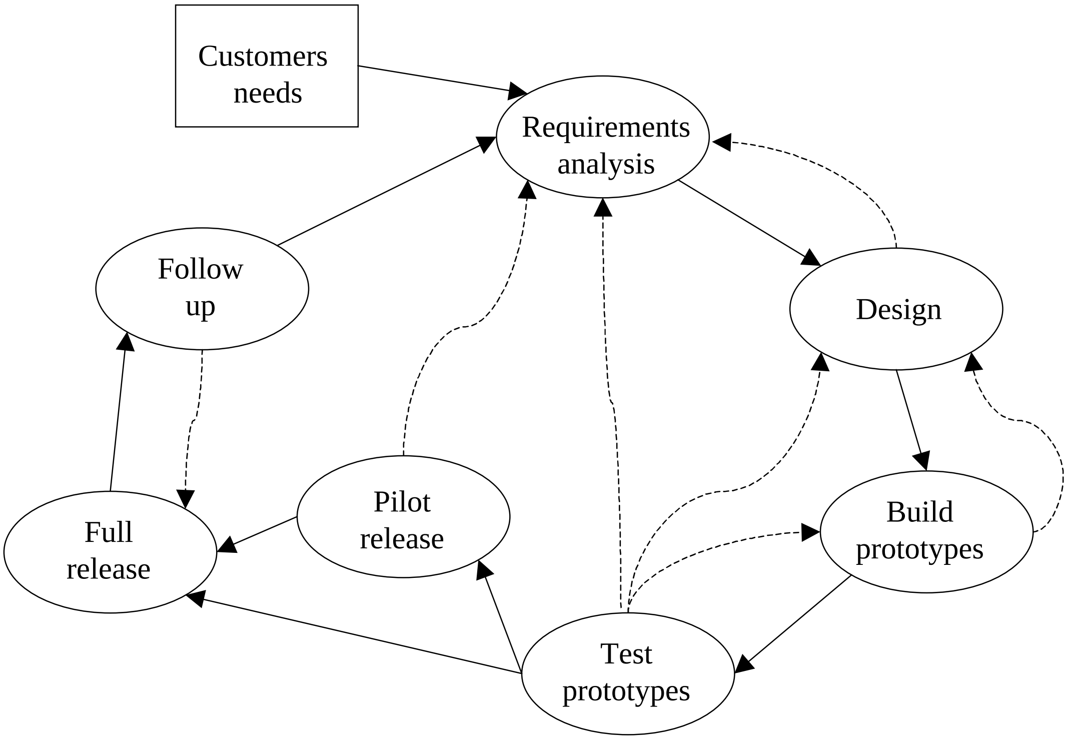


Figure 5: The Product Development Life Cycle

The System Development Life Cycle (SDLC)

A system development life cycle resembles the product development life cycle. It usually incorporates the steps of planning, analysis, design, implementation, and support [Whitten and Bentley, 1998]. Depending on the structure of the intended system, a system development life cycle may follow a structured development methodology (SDM) [Davis and Olson, 1985; Li, 1990], a rapid prototyping methodology (RPM) [Boar, 1984; Lantz, 1986; Naumann and Jenkins, 1982], or a spiral development and enhancement method (SDEM) [Boehm, 1988]. The SDM typically is applied to a system with clear requirements definitions, well structured processing and reporting, and a long and stable life expectancy. Table 2 shows a detailed breakdown of the SDM process. Under this methodology, iterations between phases in the process are strongly discouraged. It is therefore call a "water-fall" process. On the contrary, the RPM process allows and encourages such iterations. It uses very high-level development tools to quickly produce an operational prototype for users to gain hands-on experience. The prototype is then improved based on users' feedback. This process is suitable for a system of ambiguous or incomplete

Table 2. Detailed Phases of Structured Development Methodology (SDM)

PDLC Phases	SDM Phases	Phase Objectives
Requirements Analysis	Service Request/Project Viability Assessment	To initiate a project and conduct cost/benefit analysis as well as feasibility study.
	System Requirements Definition	To define project scope, analyze the existing system, and define information requirements, data attributes, and system objectives.
Design	System Design Alternatives	To identify and evaluate alternate system designs and prepare initial project schedules.
	System External Specifications	To specify data flow, user/system interface, system controls, and manual supporting procedures.
	System Internal Specifications	To specify processing logic, file structure, module interfaces, and system architecture.
Build Prototypes	Program Development	To transform programs' internal specifications into program code using a computer language.
Test Prototypes	Testing	To verify and validate the system being developed throughout the system development life cycle.
Pilot/Full Release	Conversion	To convert the data formats and procedures for the new system.
	Installation	To install the hardware and software for the new system, and cutover the system into production.
Follow Up	Post Implementation Review/Maintenance	To monitor and maintain the quality and performance of the new system.

Source: Adapted from Li [1990].

requirements with ad hoc reporting capability, or a system of a short life span with ever-changing requirements. Finally, the SDEM combines the RPM process with the SDM process to shorten the development time required by a project adopting the SDM process. More recently, the object-oriented development methodology began to replace the conventional SDM process as a new paradigm of system development.

Traditionally, the quality of software product is assured by applying software quality assurance (SQA) techniques and methods to software development process. Today merely SQA is not enough to achieve the quality of software product demanded by the customer. One must apply the TQM methods to the entire software development organization, not just the development processes themselves. Such practice is known as "software total quality management" (STQM). The STQM is a fundamental cultural change from the traditional quality perspective to an organizational philosophy that incorporates quality improvements in every aspect of the organization. Therefore, SQA provides a methodology to assure quality while STQM provides a framework to continually improve it.

According to Masaaki Imai [1986], the superordinate goal of making a profit can be achieved by increasing quality, reducing costs, and reducing schedules. By instilling TQM's continuous improvement strategy in every aspect of the software development, an organization never settles for the level that it has reached, no matter how good the product is. Continually challenging the status quo for new concepts and strategies helps to furnish the software industry with continually better products. Many software companies are starting to implement such culture into their organizations and empowering their employees with the ability to help make improvements even at entry-level positions.

Applying Deming's Fourteen Points to Software Development

Deming’s fourteen points of management approach provide guidelines for implementing the TQM concept. These fourteen points can be applied to managing software development processes. The following discussion is based on the framework of the system development life cycle presented in the last section.

1) **Create constancy of purpose for improvement of product and service**. Software development process traditionally ends when the completed system is handed over to the support group and put into production mode. Under the TQM culture, there is no finish line for the development team. Maybe there is a shift of focus from one project to another. The development team should be responsible for what they delivered, not the support group. Any quality problem occurs during the production should be addressed to the development team. Management must [Zultner, 1988]:
   - Establish operational definitions for each step in the software development process.
   - Define what is meant by “service to the customer.”
   - Define standards of development, maintenance, and service for the next year and five years ahead.
   - Define the internal and external customer.
   - Develop ways to provide better systems and services in less time, using fewer resources.
   - Invest in tools and techniques for better software development.
2) **Adopt the new philosophy of total quality**. Quality is everyone’s business. Not just the worker, management is part of the quality team. Under the TQM culture, quality comes first and everyone must join in. Corporate management, from top to bottom, must embrace the TQM concept and clearly communicate their support of this concept to all members in the software development team.
3) **Cease dependence on mass inspection to achieve quality**. Quality is built in, not added on. It is better to prevent errors in code, rather than reworking the code to remove the errors. Inspection or testing cannot prevent errors from happening, only experience and knowledge can. Management must install programs to continually improve software development processes. Examples of such programs are job training and job incentive programs.
4) **End the practice of awarding business based on price tag alone**. Many software organizations today are outsourcing their projects to subcontractors. It is important not to award a software contract based on price tag alone. Quality is more important than the difference in costs. Low quality in the long run will result in high total cost. It is better to create long-term relationship with a few loyal and trustworthy suppliers who can produce quality code for your system.
5) **Improve constantly and forever the systems of production and service**. System development processes must be constantly improved by introducing new and working methodology, paradigm, standards, practices, techniques, tools, policies, and procedures. All these require the organization to constantly keep tracking the best practice in the field of management information system (MIS)—the so-called learning organization. Each individual staff member is required to improve oneself by updating or even expanding one’s skill set.
6) **Institute training on the job**. To build quality into the software, the development team must have appropriate experience and knowledge. On-the-job training program is an effective means of obtaining such experience and knowledge. In the broadest sense, all MIS staff members must know what their jobs entail and how to do their work. Management must assess the skill level of an employee before he or she is assigned to a software project. Different skill levels can play different roles and assume different responsibilities in a project.
7) **Institute leadership.** Management must lead, not punish. It is manager's job to help MIS staff do a better job and create a better system. Project managers must be trained in basic interpersonal and analytical skills. They must have a solid understanding of statistical process control. They should know that in any software development team whose performance is in

statistical control, half of them would always be below average. They should focus on those members whose performance is out of statistical control.

8) **Drive out fear of job insecurity.** Employees must feel secure before they are willing to ask questions, make suggestions, or even expose their weaknesses by asking for help. The policy of long-term employment could easily drive out the fear of job insecurity. Moreover, any MIS staff whose performance is out of statistical control should be offered help in retraining or reassignment. However, if one consistently rejects helps from one's co-workers or supervisors, a layoff may be the last resort.

9) **Break down barriers between departments or staff areas.** Software development requires collaborative effort between users and IS staff. For as long as we can remember, communication gap has been the major factor to many MIS implementation failures. Furthermore, today's business system projects would most likely involve different functional areas and require expertise in database processing, client-server computing, and network installation, etc. Therefore, open communication among functional areas and general knowledge across disciplines are necessary for a successful system implementation. This requires appropriate education and training for team members to change their behavior and improve their knowledge.

10) **Eliminate slogans, exhortations, and targets for the workforce.** Slogans do not build quality systems. MIS management should not ask for impossible target or schedule, or unrealistic level of productivity. Instead, they should post their progress in responding to suggestions and in helping the staff improve quality. Let the employees put up their own signs and slogans [Zultner, 1988].

11) **Eliminate numerical quotas, , and work standards.** Quotas (such as metrics), goals (such as schedules), and work standards (such as unit times) address numbers, not quality. A software development project that causes haste and non-conformities accomplishes nothing and services no one. Let the project members put up their own goals. Managers should concentrate on helping people do a better job by reducing rework, errors, and waste. Everyone must work toward constant improvement, not the achievement of some arbitrary, short-term goals [Zultner, 1988].

12) **Remove barriers to pride of workmanship.** All people are motivated. They would like to make quality products. However, a good workmanship relies on good materials, good tools, good methods, and good timing. Poor materials, broken tools, ineffective methods, or belated schedule are all barriers to pride of workmanship and should be eliminated. Let the software development team put its group identity or team members' names on the software product to take the credit (or the responsibility) of their work.

13) **Institute a vigorous program of education and retraining for everyone**. On-the-job training is effective, but slow, for an employee to acquire skill set for a particular type of job. In today's MIS arena, technology is changing so fast that new skill set is needed for the same type of job in a short period of time. Management must set aside enough budgets to execute a generous education and retraining program for everyone to improve oneself. Under the TQM culture, all employees must know enough statistical method to understand the nature of variation, to manage the special causes of variation. Support for training employees to acquire necessary statistical method should be institutionalized.

14) **Put everyone to work to accomplish the transformation**. The TQM transformation is everyone's job. Everyone has a customer. Ask yourself who is the person receiving your work? All of us must identify our customers in order to determine precisely what our jobs are. Everyone belongs to a team, to work in the Plan-Do-Check-Act cycle, to address one or more specific issues, to find special causes detected by statistical signals. Moreover, we must put management to work. Only management can change the culture and environment that dominate any individual's performance. Management must agree on their meaning and on the direction to take. They must acknowledge their mistakes, if any, and have the courage to change. They must explain to a critical mass of people in the organization why change is necessary and that the change will involve everybody. Obviously, people must understand the Fourteen Points to know what to do and how to do it [Walton, 1986].

6. Conclusion and Recommendations

Total quality management is not only a philosophy of work but also an ethic of workers. It is coming from the wisdom and the teachings of many quality improvement gurus. It has helped many companies to improve quality of products and processes, and in turn, increase the productivity and the profitability. Any software organization that is planning to implement the TQM must have the critical mass of its employees embrace the TQM philosophy and methods before jumping onto the bandwagon. That is, all employees regardless of their ranks must fully understand (or be trained with) and internalize the TQM concept and tools. To increase the chance of success, a TQM-implementation project should start from the top management and unfold it downward to lower-level management and workers with a goal to benefit the critical mass of employees. Specifically, the goal is to improve the quality of work life for the employees through improving work conditions, work methods, work compensation, work relations, and providing the employees with opportunities for professional development. Only with this goal could we gain the full cooperation from the employees and bring about successful TQM implementation.

One caveat is that there is no free lunches for those who performance TQM activities. Once you implemented TQM concept and methods, you are bound to continually improve your products and processes. You must constantly ask yourself "What and how can I do it better next time?" There is no finish line for the PDCA (plan-do-check-act) wheel. Most importantly, there is no such things as few (i.e., the management) or mindless majority (i.e., the workers). Everyone related to the value chain of the product are significant and must use his or her mind constantly to play his or her own role well, otherwise, the chain will be broken, and the TQM process will soon fall apart. In other words, the number of people involved in TQM implementation would most likely determine the chance of TQM success. Therefore, it must be kept manageable, preferably no more than 30. A software development project typically has a team size less than 30 members; it lends itself to TQM implementation. For an organization having large number of people, we strongly recommend the use of modular approach to implementing TQM. Never try to transform all at once the entire complex organization into a TQM-compliant organization. Experience has told us that this is very unlikely to succeed.

REFERENCES

[1] Bemowski, K. (ed.) "The Quality Glossary," *Quality Progress*, February 1992, p. 28.

[2] Boar, B. H., *Application Prototyping: A Requirements Definition Strategy for the 80s.* New York: Wiley-lnterscience, 1984.

[3] Boehm, B. W., "A Spiral Model of Software Development and Enhancement," *Computer*, Vol. 21, No. 5 (May 1988), pp. 61-72.

[4] Brassard, M., *The Memory Jogger Plus*. Massachusetts: GOAL/QPC, 1989.

[5] Crosby, Philip B. *Quality Is Free: The Art of Making Quality Certain.* New York: McGraw-Hill, 1979.

[6] Crosby, Philip B. *The Strategy of Situation Management.* Boston, MA: Industrial Education Institute, 1969.

[7] Crosby, Philip B. *Cutting the Cost of Quality; The Defect Prevention Workbook for managers.* Boston, MA: Industrial Education Institute, 1967.

[8] Davis, G. B., and Olson, M. H. *Management Information Systems: Conceptual Foundations, Structure, and Development.* New York: McGraw-Hill, 1985.

[9] Deming, W.E., *Out of the Crisis.* Cambridge, MA: Massachusetts Institute of Technology, Center for Advanced Engineering Study, 1986.

[10] Deming, W. Edwards, *Quality, Productivity, and Competitive Position.* Cambridge, MA: Massachusetts Institute of Technology, Center for Advanced Engineering Study, 1982.

[11] Deming, W. Edwards, *On the Management of Statistical Techniques for Quality and Productivity.* Washington, D.C.: W. Edwards Deming, 1981.

[12] Department of Defense, *TQM Guide, Vol. 1, Key Features of the DOD Implementation*. DOD 5000.SIG.Final Draft 15 FEB 1991, p. 2.

[13] Department of Defense, *Total Quality Management—A Guide for Implementation*. DOD 5000.51-G, Feb. 15, 1990 (Final Draft).

[14] Feigenbaum, Armand V. *Total Quality Control: Engineering and Management.* New York, McGraw-Hill, 1961.

[15] Feigenbaum, Armand V. "The Challenge of Total Quality Control," *Industrial Quality Control*, May, 1957, pp. 17-23.

[16] Feigenbaum, Armand V. *Quality control: principles, practice and administration.* New York, McGraw-Hill, 1951.

[17] Imai, Masaaki, *Kaizen: The Key to Japan's Competitive Success*. New York: McGraw-Hill, 1986.

[18] Ishikawa, Kaoru, *What Is Total Quality Control? The Japanese Way.* [English translation by David J. Lu.] Englewood Cliffs, N.J.: Prentice-Hall, c1985.

[19] Ishikawa, Kaoru, *Guide to Quality Control.* Tokyo: Asian Productivity Organization, 1976.

[20] Juran, Joseph M. *Quality-Control Handbook.* 1st ed., New York: McGraw-Hill, 1951.

[21] Juran, Joseph M. *Management of Inspection and Quality Control.* 1st ed., New York: Harper, 1945.

[22] Lantz, K. E., *The Prototyping Methodology*. Englewood Cliffs, NJ: Prentice-Hall, 1986.

[23] Li, E.Y., " Software Testing in A System

Development Process: A Life Cycle Perspective," *Journal of Systems Management*, Vol. 41, No. 8 (August 1990), pp. 23-31

[24] Moore, W.S., "Singing the Same 'Total Quality' Song, " *National Defense*, March 1990, pp. 29-32.

ABOUT THE AUTHORS:

Eldon Y. Li is Professor and a former Coordinator of MIS area at the College of Business, California Polytechnic State University, San Luis Obispo, California, U.S.A. He is currently visiting the Department of Decision Sciences and Managerial Economics at the Chinese University of Hong Kong. He holds a bachelor degree from National Chengchi University in Taiwan and M.S. and Ph.D. degrees from Texas Tech University. He has provided consulting services to many firms for a variety of software projects and served as a management consultant to the clientele of the U.S. Small Business Administration. He was a software quality specialist at Bechtel Corporation's Information Services Division and a visiting software scientist at IBM Corporation. He is a Certified Data Educator (CDE) and is Certified in Production and Inventory Management (CPIM). His current research interest lies in human factors in information technology (IT), strategic IT planning, software engineering, quality assurance, and information and systems management.

Houn-Gee Chen is Professor and Chair of Department of Information Management at National Chung Cheng University (NCCU) in Taiwan. He holds a bachelor degree from National Tsing Hua University and MS and Ph.D. degrees from University of Wisconsin-Madison. Before joining the NCCU in Taiwan, he held a faculty position at the University of Notre Dame. His current research interest lies in decision support systems, group technology, information technology management, software engineering, and quality management.

Waiman Cheung holds an MBA and a Ph.D. in Decision Sciences and Engineering Systems from Rensselaer Polytechnic Institute. He is currently an Associate Professor of the Business Faculty, The Chinese University of Hong Kong where he teaches both graduate and undergraduate level MIS courses. Prior to that he had operated his own MIS consulting company and worked as a Technical Staff for Oracle Systems in the US. His research interests include Information Systems Integration, Database Modeling, Electronic Commerce, and Data Mining.