Transactions on Information and Communications Technologies vol 4, © 1993 WIT Press, www.witpress.com, ISSN 1743-3517

Japanese approaches to software quality management

M. Bush[a], J. Hemsley[b]
[a]Centre for Systems & Software Engineering, School of Computing, Information Systems and Mathematics, South Bank University, Borough Road, London SE1 0AA, UK
[b]Brameur Limited, 237 High Street, Aldershot, Hampshire, GU11 1TJ, UK

ABSTRACT

This paper gives an overview of the typical Japanese approaches to software quality management, concentrating on the use of measurement. It begins by briefly reviewing Japan's history and cultural make-up, since most analysts are agreed that differences in culture are the main reason for differences in industrial practices between Japan and 'the West'. The paper then gives an overview of the Japanese software industry in general terms, leading to a discussion of the use of measurement within it. The Japanese computer manufacturer NEC is included as a case study. The paper ends with a brief description of some relevant work reported at a recent international conference.

BACKGROUND

In industrial terms Japan is an acknowledged world leader. In particular, their ability to mass produce high quality, low cost, high technology goods such as automobiles and consumer electronics is second to none.

Numerous studies in recent years have attempted to analyse the way Japanese industry operates and to draw lessons from its success. While the underlying reasons for it are many and varied, the conclusions of such studies have invariably highlighted cultural differences between Japan and 'the West' as being of vital importance. ('The West' here means western Europe and north America, and other nations such as Canada and Australia which have a similar cultural heritage.)

In order to understand the Japanese way of working it is essential to appreciate the basis of their culture, which gives rise, generally speaking, to rather different attitudes towards work than are usual in the West. Whilst it is always important to regard generalisations with a degree of scepticism, they are reasonably dependable when applied to Japan because the 120 million or so Japanese are a fairly uniform people. Although Tokyo ranks alongside New York and London as one of the biggest and most developed cities in the world, it does not have anything like the multi-cultural atmosphere of its counterparts.

This paper therefore begins by discussing the basis of Japanese history, culture and industry in general before focussing in on the software industry and its use of measurement in particular.

History (Since 1637)

During the Edo period, after the Shimabara uprising in 1637, Japan was almost completely sealed off from the rest of the world. The tyrannical rulers forbade anyone to enter or leave Japan. The country remained almost totally isolated for over 200 years; cut off from international debate and progress in areas such as science and technology. Society in Japan was highly organised, with clearly defined classes - nobility, military (Samurai), farmers, craftsmen and merchants. Personal conduct and even dress was dictated by detailed laws and regulations, which were brutally enforced by the Samurai. The masses learned to obey orders without question, and it seems that a willingness to follow orders still remains in the national psyche.

Japan's isolationism was abruptly brought to an end in 1853 when a powerful fleet of the US navy paraded itself invincibly on Japan's shore-lines and demanded that Japan opened its borders to trade. Lack of cooperation by the Japanese led to a three-day bombardment by British, American, French and Dutch ships in 1864. This persuaded the Japanese once and for all that they needed to open up and modernise.

Although the Samurai were harsh rulers, compared to earlier times the Edo period provided an atmosphere of relative peace and stability, in which the merchants were able to flourish. Their influence within society steadily grew, and they effectively laid the foundations for Japan's future commercial success. The pace of development once Japan opened up was astonishing. Within 50 years Japan had transformed itself from the status of a feudal agricultural nation to one of the world's most powerful and dynamic democratic societies, with a modern army and navy, industry of all kinds, a network of railways and a parliament.

However, the newly awakened national spirit, and a growing frustration brought on by the unwillingness of its Asian neighbours to buy its industrial products, led Japan into a succession of conflicts and conquests of its neighbours. Japan's aggression finally led to their involvement in the second world war, in which much of Japan was destroyed and Hiroshima and Nagasaki became the first and only (to this day) cities to have ever been devastated by atomic bombs.

This was in fact the very first time in recorded history that Japan had been conquered by an invader. Although Japan suffered severe damage and economic set-back, the defeat precipitated a second industrial transformation in Japan which, like the first one 80 or so years earlier, has resulted in quite amazing industrial and economic growth. Japan has now reached the status of economic superpower. Its per capita assets are three times those of the USA and four times those of the UK. Personal savings rates are currently, on average, around 14% of income; the highest in the world.

Transactions on Information and Communications Technologies vol 4, © 1993 WIT Press, www.witpress.com, ISSN 1743-3517

Modern Industrial History & Culture

Undoubtedly an important force in Japan has been, as in the former West Germany, a national desire to regain status after the second world war. At least this has been the outlook of both countries' leaders, if not their whole populations. Since the war both countries have focussed their efforts on building up their commercial industries; military spending was suppressed by conditions imposed by the allies at the end of the war. This has given them a considerable advantage over countries such as the UK, France and the US which have maintained much higher levels of military spending. For example, Japanese companies now operate using very high technology platforms as a result of on-going investment. More industrial robots and office-automation equipment are to be found in Japan than in any other country.

Research and development in Japan tends to be company-led, rather than government-led. The large companies often work together on major R&D projects. They also cooperate on the development of standards which are subsequently adopted nationally. In the West things tend to work the other way around; standards are proposed by government bodies often working together with academic institutions, and only then are companies encouraged to adopt them.

Industrial success for the Japanese has come largely from their long-standing commitment to, and now their reputation for, high quality. This commitment can be traced back to the early years after the war when several industrial 'gurus' from the USA went to Japan to act as advisers in the rebuilding of Japanese industry. The most famous of these was W. Edwards Deming, whose management principles are still much honoured.

Many companies in Japan operate as part of 'business families'. Before the second world war these were known as Zaibatsu. A typical Zaibatsu consisted of a tightly coupled collection of key manufacturing and service organisations under the influence of one major shareholding company. The chairmen of the largest Zaibatsu were very influential members of society. However, all this changed after the war. The Zaibatsu were dissolved and new laws were put into place effectively preventing individual members of society acquiring too much personal wealth and influence.

Instead, some of the groups of companies which had originally formed Zaibatsu got together to form business families of a rather different kind. The idea was still that these companies would support each other as much as possible through trade and servicing agreements, but now the companies were free to operate in their own best interests. Bonds between the companies were usually reinforced by a small amount of mutual shareholding.

Many of these business families are still in operation, and indeed some are very successful. Often such a family will include a bank, a computer/electronics company, a pharmaceutical company and other key manufacturing and service organisations. Well known examples are the Mitsubishi, Sumitomo, Furukawa and Mitsui groups. Decision making within these families is distributed amongst the member companies. By contrast, in the Zaibatsu the decision making power was centralised.

Transactions on Information and Communications Technologies vol 4, © 1993 WIT Press, www.witpress.com, ISSN 1743-3517

In other cases smaller companies cluster around one giant; this is the case with Toyota, Honda and Hitachi. Such clusters are known as Keiretsu. Although similar in structure to Zaibatsu, important differences are the fact that the central giant is always a major producer of goods or services, and itself must be under the directorship of a board of directors. Business empires of the kind controlled by Robert Maxwell or Richard Branson are not to be found in post-war Japan.

In general, Japanese companies have long-term rather than short-term ambitions, and are therefore inclined towards increasing their market share rather than increasing short-term profits. It is this attitude which often leads to Japanese products being exported at very low profit margins, much to the distaste of foreign governments who see these cheap imports as a threat to their national manufacturing industries. In many cases Japanese products are actually more expensive to buy in Japan than they are in other countries. Virtually all the profits which are made are invested in areas such as research and development, and in improving the manufacturing process. Most Japanese companies seem to be continually and self-consciously striving for improvement.

Despite several recent financial scandals and allegations of corruption, most business in Japan is sincere, open and ethical. There is usually a good deal of trust involved in business deals, with few litigation proceedings. Consequently the Japanese have a rather relaxed attitude towards written contracts. This has particular implications for the software industry, as we shall see.

The Education System

The stresses which the education system in Japan applies to Japanese children are well known. Children are taught early on, at home and at school, to respect and obey their elders, to work with each other in groups, and above all to work hard. The ultimate aim for most students is to gain entry to a university; the more prestigious the university the better.

Entry as a professional engineer into a prestigious company, for example, depends solely on achieving the appropriate level of school and then university education. Competition for places at good universities is extremely high, since it is very difficult in Japan to catch up later after a poor academic start in terms of a career.

The Japanese education system is very successful in the sense that it produces a high proportion of very well educated young adults. Teachers tend to concentrate on eliminating weaknesses in the individual rather than nurturing strengths, which is often the focus of Western education, and in most cases rote learning is valued above creativity. Children are subjected to repeated testing and grading, and there are also ranking systems for schools and universities. The Japanese themselves admit, however, that there are significant psychological and sociological problems associated with the style of education most children receive, and particularly with the fact that so many children are put under so much pressure to achieve good results.

Transactions on Information and Communications Technologies vol 4, © 1993 WIT Press, www.witpress.com, ISSN 1743-3517

Although illiteracy in Japan is virtually non-existent, and roughly 40% of school leavers go on to higher education, the problem this creates is that overall the work force is over qualified. There are not enough people willing to do the more manual jobs, such as work on construction sites. These jobs are increasingly being filled by illegal workers from poorer Asian and Middle Eastern countries, such as Bangladesh, Iran and the Phillipines. Everybody knows this is happening, but the authorities seem to be quite tolerant about it.

Employer/Employee Relationships & Attitudes

Although Japan is now in many senses a 'Westernised' society, with countless discos, fast food restaurants and amusement arcades, their way of life, and to a certain extent their outlook on life, is still distinctly different. Whereas the principles and philosophies of the West are derived from such influences as Judeo-Christianity, Greek rational thought and Roman administrative systems, those of Japan stem from Buddhism, Shintoism and Confucianism. One of the main differences which arises is that whereas, deep down, most Westerners regard themselves as true individuals who 'only live once', most Japanese tend to regard themselves fundamentally as part of the society in which they live, which, as in Confucianism, is thought to be the greatest good.

Generally speaking, the Japanese can be said to have a tendency towards:

- respect for elders, and for seniority within society
- respect for ethics and morals
- willingness to conform
- perfectionism
- diligence
- team working
- competitiveness
- long-term thinking.

The picture of the stereo-typical Japanese male industrial worker is well known. He is devoted to one company, never having worked for any other, and lives with his family in the same neighbourhood as many of his colleagues, perhaps in a company-owned apartment. The school his children attend is probably also supported by the company. His family often joins those of his colleagues on company holiday excursions. In these respects Japanese society is almost communist in nature.

The situation is changing, though. Whereas before the second world war many Japanese may genuinely have conducted their lives for the good of society as they saw it, now they are no doubt mostly motivated simply by wanting to do well for themselves and their immediate families. To that extent they have become more Westernised; the ideals of Buddhism and Confucianism have definitely begun to wane.

Although personal income is roughly the same as it is in the USA or the former West Germany, Japanese employees have to work an average of 2,160 hours a year to earn it, compared with 1,980 in the USA and only 1,640 in the former West Germany. Working long hours seems to be more socially acceptable than it is in the West; anyone who regularly leaves early (that is, on time) risks being viewed as a shirker.

Staff turnover is extremely low in Japan, and this has a number of distinct advantages over the much higher turnover rates common in the West. For example:

- workers are likely to take more care over their work, perhaps because they feel a stronger sense of loyalty to the company, and because they are more likely to still be around in the future to face the consequences of any shoddy work
- communication and cooperation between co-workers is likely to be more effective since they will have worked together for many more years, on average, and are quite likely to be social acquaintances
- administrative and other systems work more smoothly on the whole because everyone is more familiar with the procedures, through having lived with them for longer.

The most successful Japanese companies appear to operate at a level of harmony which few Western companies seem to be able to match. No doubt the factors just mentioned have a lot to do with this. By contrast, there is often tension within many companies in the West caused by conflict between different departments, and by repeated changes in company structure brought about by changes in upper management. Newly appointed managers in the West too often want to 'make their mark' by reorganising existing structures or procedures. This usually creates a considerable amount of upheaval, and although probably well intentioned, may well cause matters to get worse rather than better. Their replacements in turn create more upheaval, and those lower down the company hierarchy who may have lived through several such upheavals can easily become disillusioned.

In return for a high level of commitment from its employees, most Japanese companies offer extremely good employee training and care programmes. In times of difficulty Japanese companies make every effort to re-train and re-deploy staff if at all possible, rather than making staff redundant. A good example of this is Nippon Steel Corporation, which in recent years has re-trained around 800 steel workers, with an average age of 43, to become software engineers. The motivation was simple; on one hand the steel industry was in a slump and on the other hand they were in need of software engineers. In similar circumstances, Western companies would be much more likely to make 800 workers redundant and independently recruit 800 qualified software engineers.

Volunteer steel workers at Nippon Steel Corporation underwent one year of full time training, after which they became junior programmers. The drop out rate was only about 10%, and apparently most of them have become competent software engineers. It is claimed that part of the reason for their success is that they are well used to working closely together in tight knit teams, and that this is an important quality when it comes to software engineering.

This kind of training is not uncommon. It is quite normal for companies to provide new recruits with several months full time training before they are expected to produce any real work. No European or American company can

Transactions on Information and Communications Technologies vol 4, © 1993 WIT Press, www.witpress.com, ISSN 1743-3517

afford this kind of training because staff turnover is so high by comparison; in fact turnover is particularly high for employees in their early working lives. It is also normal in Japan for newly appointed managers to receive two or more weeks of full time intensive training in management techniques, with additional intensive training each time they move a step up the management ladder.

The vital importance of good motivation is widely recognised in Japan as the key to getting the best out of people. This is especially important in the software industry where employee costs far outweigh all other costs, and the performance of individuals can be so variable. Also, Japan has a long tradition of excellence in craftsmanship, and most Japanese workers do seem to take considerable pride in their work. As customers they also expect the best; the discerning nature of the average customer in Japan was the original motivator which caused Japanese companies to concentrate on producing products of high quality.

THE SOFTWARE INDUSTRY IN JAPAN

Although the software industry in Japan has not yet reached the high levels of success attained by other industries, there are signs that it may do in the near future. If this happens it is likely to be because of the consistent achievement of high quality. Most of the successful Japanese software producers seem to recognise that productivity in itself is a poor goal since it tends to lead to poor quality, and prolonged testing and debugging activities. As in other areas of industry, the achievement of high quality is the driving force.

Industry Structure

The two main software purchasing bodies in Japan, in the absence of any significant defense spending, are the computer manufacturers, who mainly need systems software, and the user community, who mainly need applications software.

There is a range of software producers in Japan. Firstly, many large companies such as Japan Air Lines, as well as the major computer manufacturers themselves, produce much of their own software. Secondly, there are about 600 companies which are affiliated to one or other parent company that acts as the sole or principal customer. Many of these are 'spin-offs' of the parent company, set up specifically to undertake software development and maintenance. Thirdly, there are about 50 truly independent software houses that do in-house development for a range of clients. Finally, there are another 5,000 or more small independent companies that act as 'body shops', hiring out engineers to work on projects based at customer sites. Contracts originating from the major customers (such as computer manufacturers, banks and transport operators) often filter down through their own affiliated companies to independent software houses in the form of sub-contracts. These may in turn hire additional body shop personnel to help them out on a short term basis.

Software development practices vary a great deal throughout the industry. The most sophisticated software producers are the large computer manufacturers and their affiliated companies, the most important being Hitachi,

Transactions on Information and Communications Technologies vol 4, © 1993 WIT Press, www.witpress.com, ISSN 1743-3517

Fujitsu, Mitsubishi, NEC and Toshiba. In addition NTT, the recently privatised national telecommunications company (and now the richest company in the world by share value), is amongst the leaders. These are the companies which have pioneered the development of 'software factories' and other state-of-the-art practices such as the systematic use of measurement, as we shall see. Partly due to the fact that these high-profile companies have first pick of new recruits, and because very few of their employees ever leave to join other companies, there is a considerable gap in ability and effectiveness between these leading software producers and the rest.

Differences in Software Production Between Japan & The West

In 1986 the total revenue for software in Japan was $5 billion, compared with $19 billion in the USA. However, as much as 96% of the Japanese software was designed for integrated software and hardware systems, in comparison with nearly 60% of the US software which consisted of mass market packages. This may go some way to account for one of the major apparent differences between Japan and the West, which is that the large Japanese firms on the whole are more concerned with process improvement while Western firms are often more concerned with product innovation. Japanese software factories try wherever possible to tackle projects which are similar to ones they have tackled before. Novel projects are usually tackled outside the factory environment. This may also account for the apparently higher figures for reused software in Japanese products.

Japanese software products on the whole seem to be more reliable than those produced in the West - figures vary, but typically indicate that Japanese software products have something like 50% fewer faults. One consequence of this is that maintenance costs are greatly reduced. Demand for software within the Japanese domestic market continues to exceed the supply capabilities of the software producers, and this has been one of the main reasons why the Japanese have not exported much software or bid frequently for overseas contracts. It is dangerous to assume that, because they do not currently export much software, their software industry is immature. On the other hand, producing software products for foreign markets is considerably harder than producing the same product for the home market because of the language barrier.

On the whole, there does not seem to be a major difference between software development practices in Japan, Europe and the USA, although minor differences certainly exist. For one thing, the low turnover rate in Japan has allowed them to rely less on documentation, since the original designers are usually still around when it comes to maintenance. Another factor here is that, due to the complexities of the written language, Japanese word processors have only been developed relatively recently.

As in other areas of business, there is also less emphasis on written contracts in Japan; Japanese companies are used to working on the basis of trust. Litigation proceedings are seen as bringing shame upon both supplier and customer. Consequently there is nothing like the interest in third party testing or certification that there is in the West where, particularly with the opening up of

Transactions on Information and Communications Technologies vol 4, © 1993 WIT Press, www.witpress.com, ISSN 1743-3517

free trade between European countries, certification is seen as providing some kind of guarantee of product quality.

Although the Japanese have considerable experience at producing very complex systems such as large-scale telecommunication systems and sophisticated computer operating systems, they have rarely pushed forward the state-of-the-art of software technology. Also, rather than develop new production techniques, the Japanese have very often chosen to copy and refine the most promising approaches used in the West. According to Cusumano, the notion of a software factory originated not in Japan but in the USA in the 1960's [1]. IBM, SDC and TRW in the USA all tried out software factory concepts, although SDC were the only ones to use the 'factory' label. However, while SDC allowed its factory experiment to come to an end in 1979 after teething problems, the Japanese factories, through greater perseverance, appear to be going from strength to strength.

One reason why the Japanese were so keen to copy was that IBM had over 50% of the world market share of computers in the 1970's. Fujitsu and Hitachi in particular decided that their best strategy would be to produce machines with IBM-compatible operating systems in the hope that users would see fit to migrate their existing software. This led to drawn out legal disputes with IBM. A settlement was finally reached between Fujitsu and IBM in 1988, lasting ten years, which essentially gave Fujitsu the right to continue copying IBM machines after paying a lump sum of several hundred million dollars, plus an annual licensing fee.

The idea that Japanese companies have become successful simply by copying American products and concepts is, however, far from the whole truth. Companies such as NEC and Toshiba chose to take a different path and have developed entirely new operating systems. In fact there have been numerous American visitors to the major Japanese software producers since the 1970's who have tried to emulate at home aspects of what they saw there. Of course, a major reason why it is the Japanese who have been able to do most of the copying is that they are so familiar with English. If only Westerners could read Japanese books, reports and research papers with the fluency that many Japanese can read English ones, the situation would surely be quite different.

<u>The Software Factory Approach</u>
The term 'software factory' became popular in Japan in the 1970's, and was adopted by several Japanese companies. Hitachi were the first to use the term, in 1969. Two factors which prompted Hitachi to set up their software factory were a severe shortage of skilled software engineers and a poor past reputation for quality. Hitachi set two goals for their newly created software factory:

- to improve productivity and reliability through process standardisation and control
- to transform software production from an unstructured service to one which could deliver products with a guaranteed level of quality.

These goals are just as appropriate today as they were then. In order to achieve them, a software factory must have a standardised life-cycle model and

Transactions on Information and Communications Technologies vol 4, © 1993 WIT Press, www.witpress.com, ISSN 1743-3517

project management procedures, and a uniform development environment utilising CASE (Computer Aided Software Engineering) tools as far as possible. It must also have a well developed data collection and analysis programme so that quality and productivity trends can be tracked and new working methods properly assessed. The notion of a software factory also places an emphasis on the use of reusable code modules.

Although strictly speaking the term 'software factory' refers to a concept rather than a physical building, often it does refer to a particular location where lots of people are engaged in software production. Typically, a software factory consists of large open plan offices with 100 or more software developers in each. Workspace is often very limited, particularly in urban areas where land prices are high. Each engineer normally has a small area of desk space, a computer terminal, and shelf and some filing space. Desks are often set out in rows, with group leaders in a position to see all the members of their group. It is also common for the department manager and his or her deputy to sit in partitioned rooms in a corner of the office.

Of course, simply adopting the factory title does not necessarily mean a different way of working, and, in general, the tools and techniques used in Japan (whether in software factories or not) are not very different to those used in the USA and Europe. The review or walkthrough, for example, is still the principal technique used for improving the quality of software. However, most Japanese software factories have launched genuine long-term efforts to centralise and systematise their software production.

Hitachi tried to run its software factory along the same lines as its hardware factories. As in other companies, the drive came from upper management rather than the software engineers themselves, who tended to dislike the idea at first, feeling that they were white collar workers being asked to operate along the same lines as blue collar workers. During the mid 1970's there was heavy investment in the development of automated tools for project management, design support, testing, program generation and reuse support. In 1985 Hitachi split its software operations into 'basic' and 'applied', since it realised that different tools, techniques and methods were needed for each.

In 1976 NEC established no less than 5 software factories, with between 1,250 and 2,500 software engineers in each one. Since then Toshiba, Fujitsu, NTT and Mitsubishi have all followed suit. (Interestingly, Hitachi have recently changed the name of their software factory to 'Software Development Centre' because they feel the factory image is actually off-putting to prospective new recruits!)

USE OF SOFTWARE ENGINEERING MEASUREMENT IN JAPAN

One area where the Japanese appear to be ahead is in the use of software engineering measurement. Nearly all the leading software producing companies in Japan are active in this area. Papers in the area recently appearing in journals or presented at conferences, for example, have originated from all of the following companies: IBM Japan, Fujitsu, Hitachi, JSD, Mitsubishi, NEC,

Transactions on Information and Communications Technologies vol 4, © 1993 WIT Press, www.witpress.com, ISSN 1743-3517

NTT and Toshiba. All of these companies can quote yearly figures for Lines-Of-Code, proportions of re-used code, and average product defect densities going back many years. The final section of this paper provides further ample evidence of recent Japanese activity in this area.

In some Japanese companies the work of individuals is even measured, something which is normally seen as quite unacceptable in the West. (In fact in Germany it is even illegal to measure the work of an individual as such.) However, workers seem to accept it on the basis that it gives managers an insight into training requirements. This is crucial; measurement of the individual is carried out with a view to *helping* rather than *penalising*.

The way quality assurance is normally tackled within the Japanese software industry is rather different from the way it is normally tackled in the West. In Japan every member of staff is encouraged to be involved with quality issues, whereas in the West there is often a detached quality team. Also in Japan quality assurance programmes are often applied company-wide rather than just project-wide, as is more usual in the West. Open relationships between suppliers and customers in Japan mean that in many cases customers are given a high degree of visibility into how projects are progressing. For example, customers may be shown graphs showing the trends in defect detection rates during various phases of the life-cycle; something which few Western suppliers would contemplate.

Japan is also at the forefront when it comes to international standards. In particular, they led the development of a new international standard concerned with software quality, ISO9126, which was published in December 1991 [2]. This standard identifies 6 sub-attributes of quality; functionality, reliability, usability, efficiency, maintainability and portability. For each of these attributes ISO9126 provides a definition and a set of suggested sub-attributes. For example, it suggests the following sub-attributes of functionality; suitability, accuracy, interoperability, compliance and security. Although ISO9126 discusses the measurement of these attributes, it recognises that "the state of the art at present does not permit standardization in this area". Nevertheless, companies such as NEC do indeed have internal standardised quality models and ways of measuring many of the attributes which form part of these models.

A Case Study - NEC

NEC (Nippon Electric Company) was established in 1899, and claims to have produced the first commercial transistorised computer anywhere in the world, in 1958. In the late 1980's NEC was Japan's largest supplier of small computers, with over 18,000 software personnel divided roughly equally between divisions of the parent company and 25 subsidiary companies.

As in other Japanese companies, new recruits to NEC receive several months initial training, with 'on the job' training and periodic short courses thereafter. As well as learning about technical subjects, new recruits are also initiated into the company philosophy and ways of working. Employees of NEC are taught to recognise four basic principles:

- every individual should motivate themselves to produce work of the highest quality

- quality should arise through sound production processes rather than through testing/debugging
- it is essential to pursue the cause of errors, although individuals should be free of blame
- error data should be fully utilised.

By contrast, few European or American companies have recognisable company philosophies at all. Rather, ways of working evolve as new managers take the helm. NEC's philosophy is clearly centred on the issue of quality, and in this respect NEC is typical of Japanese companies. Quality assurance within NEC is achieved through having around 1,600 'Quality Circles'. A quality circle is a group of usually around 5 or 6 people dedicated to monitoring and improving quality. The idea of having quality circles in software development was first promoted company-wide in 1980, although they had already been popular for some time in the area of hardware manufacturing. On a large project there will be several such teams, each dealing with a different part of the code. Each quality circle is expected to meet once a week during working hours. (In Western companies such meetings, where they occur, are often squeezed into lunch hours.)

NEC hold two annual company conferences on quality, mainly as a motivational exercise. Typically, the 1,600 quality circles will between them submit over 1,000 papers, of which about 100 are selected for presentation. Of these, perhaps 30 will be awarded prizes. The chairman and other senior company executives are nearly always present at these conferences to demonstrate that the issue of quality is taken seriously at the very highest levels.

NEC has been making use of software engineering measurement for many years. Having started to operate a software engineering measurement programme in the late 1970's, they recorded the following cost improvements by the mid 1980's:

Cost of coding -	increased 10%
Cost of debugging -	decreased 30%
Cost of testing -	decreased 50%
Cost of maintenance -	decreased 67%.

Although nobody claims that these improvements came about *because* of the use of measurement, knowing the figures at least allowed NEC to judge the impact of other changes in their working practices. For example, NEC were able to confirm their hypothesis that software which was written *clearly* was likely to be of high quality, without penalty in terms of productivity. As a result they now stress clarity as an important goal in software production.

Most of the measures NEC use are concerned with the processes of producing and later of running the software, rather than with the software itself. Current projects undertaken within NEC typically collect the following process data:

Process	*Data Collected*
Planning	Estimated program size;
	Estimated effort requirements;

Transactions on Information and Communications Technologies vol 4, © 1993 WIT Press, www.witpress.com, ISSN 1743-3517

	Scheduled completion date; Development staff personnel; Member experience in years; Potential bug count; The target number of bugs to detect; Development difficulty; Degree of market need; Development form; Language used.
Basic, functional & detailed designs	Process completion date; Actual effort required; Number of specification pages; Review effort; Bug detection count; Number of test cases designed; Estimated program size; Progress information.
Coding	Process completion date; Actual effort required; Actual program size; Inspection effort; Bug detection count; Number of test cases designed; Progress information.
Testing	Process completion date; Actual effort required; Scheduled number of test cases; Target number of bugs to detect; Number of test cases executed; Bug detection count; Progress information.
Maintenance	Field trouble count; Maintenance effort.

NEC's goals are essentially to assess productivity, quality and delivery overruns. These are assessed by evaluating a number of 'standard' ratios, such as:

- *Overall Productivity* = Lines Of Code ÷ Effort
- *Reuse Level* = Reused Lines Of Code ÷ Effort
- *Density Of Bugs Detected Before Release* = Number Of Bugs Detected Before Release ÷ Non-Reused Lines Of Code
- *Review Level* = Review Effort ÷ Non-Reused Lines Of Code
- *Bug Detection Rate In Upper Process* (ie. Design & Coding) = Number Of Detected Bugs In Upper Process ÷ Review Effort
- *Bug Detection Density In Upper Process* = Number Of Detected Bugs In Upper Process ÷ Non-Reused Lines Of Code
- *Average Bug Detectability* = Number Of Bugs Detected ÷ Total Effort Expended On Detection
- *Delayed Component Rate* = Number Of Components Delayed ÷ Total Number Of Components.

Transactions on Information and Communications Technologies vol 4, © 1993 WIT Press, www.witpress.com, ISSN 1743-3517

Using figures such as these, collected by many projects over successive years, NEC are able to plot graphs that clearly show which areas are improving and which are not. Furthermore, senior managers are able to compare the strengths and weaknesses of different projects. Such quantitative figures also give real insight into the progress of projects which are still on-going.

A BRIEF REVIEW OF SOME RECENTLY REPORTED WORK

This final section briefly describes some relevant work which was reported in November 1992 at the European Organisation for Quality (EOQ) Conference on Software Quality. The Japanese sent nearly 40 delegates comprising the *third* largest national delegation to this conference, and made a correspondingly strong contribution in the papers presented.

Professor Kanno from the Science University of Tokyo, the leader of the Japanese delegation, presented the first Japanese paper in a plenary session entitled 'Towards Excellence'; his paper was entitled 'Japanese Software Quality and the Deming Award'. It covered three main topics:

- an overview of Japanese Software Quality activities including the 'House of Total Control'
- the Deming prize
- a proposed approach for 'International Software Quality Assurance from the Viewpoint of World-Wide High Quality C, C & C' (*C, C & C = Control, Computing & Communication*).

Professor Kanno's paper highlighted the Japanese emphasis on behavioural aspects of quality assurance, which contrasted with the approaches in subsequent papers from the Software Engineering Institute in the USA and the new British IMPROVE IT initiative which is supported by the Ministry of Defence.

A paper entitled 'Software Development Management based on Process Curves Categorisation' by Yasuhiro Kadota and three colleagues from NEC TELECOM SYSTEMS described a management method method based on categorising project progress curves in six basic patterns:

- delayed start
- increase in delay of progress
- a 'plateau' due to work interruption
- negative progress due to re-work
- 'stagnancy' due to work difficulty, especially at the '90%' level
- slow progress at earlier stages.

By analysing the causes and identifying appropriate counter-measures for each type of progress problem, NEC TELECOM have been able to enhance their *preventative* quality assurance system. The methodology appears to be a very pragmatic approach.

Transactions on Information and Communications Technologies vol 4, © 1993 WIT Press, www.witpress.com, ISSN 1743-3517

Another paper, by Heideku Nogi of the Computer Institute of Japan, entitled 'The Success and Failure of Software Projects and Their Productivity', presented a quantitative analysis of 33 projects. The author's chief conclusion is that "there is almost no relationship between the success of or failure of projects and productivity". However, the results indicate the importance of domain complexity and the development environment.

The Japanese played a leading role in the session on Quality Function Deployment, with two papers being presented. The first, by Dr Ohmori of Fujitsu entitled 'Framework Design of Software Quality Deployment', states that, despite efforts to apply the Quality Deployment Approach (QDA) over the last ten years, "a framework of QDA for software is still immature". He discusses how a framework for QDA should be designed, citing the critical issues as:

- clarification of basic standpoints for design
- making conceptual and methodological designs.

The focus of the paper is on business software but Dr Ohmori emphasises that the approach should also be relevant for other types of software.

The other paper in the session, by Professor Kooro entitled 'Structural Way of Thinking as Applied to Increasing Customer Satisfaction', presents a fascinating combination of methodological and behavioural issues. The importance of the latter in Professor Kooro's eyes is shown by such statements as: "The leader must be customer conscious and show strong leadership" and "An important key is found in the history, when such an organisation begins to succeed". Professor Kooro also offers a refreshing insight on the state-of-the-art in Japan on the topic of requirements descriptions; "... these chaotic documents are valuable to know customer needs, but a more systematic and better way of describing requirements is needed".

A paper by Keiko Koga of Hitachi, entitled 'Software Reliability Design Method in Hitachi', provides a description of their approach to raising reliability levels based on twenty years of experience. Their method involves reliability target setting based on statistical analysis of previous projects. The level of errors detected during the various test stages is then used for management control purposes. The approach is based on over 100 projects and is judged as particularly successful for large projects, although Hitachi are continuing to refine and improve their methods, for example by paying further attention to human factors and differing types of development conditions.

Finally, a paper by Fumiaki Teshima and Raymond Jacoby of Toshiba entitled 'Macroscopic and Microscopic Approach for Quality Assurance of Software Systems' discusses differing approaches to modelling software reliability. Whilst the "macroscopic approach" includes the classic software reliability estimation models such as the Gompertz Growth Curve, the "microscopic approach" is based on a statistical failure model which provides an estimate for the total number of failures for systems as a function of parameters such as:

- problem difficulty
- sufficiency of testing

Transactions on Information and Communications Technologies vol 4, © 1993 WIT Press, www.witpress.com, ISSN 1743-3517

- ability of developers
- ability of testers
- characterisation of the development environment.

An estimate selection method is proposed by the authors based on fuzzy clustering techniques. This approach then leads to proposals for practical introduction of the proposed approaches with a reliability estimation tool consisting of two components:

- a reliability estimation component
- a reliability selection component.

Following case study analysis, the authors conclude that the "macroscopic approach" is indeed more promising for further research.

ACKNOWLEDGEMENTS

The authors would like to acknowledge the support of the European Commission who, through the two ESPRIT projects METKIT and PYRAMID, sponsored a visit (by the authors and two others) to a number of Japanese companies in September 1991.

The authors would also like to thank Professor Motoei Azuma of Waseda University, previously a senior manager at NEC and chair of the ISO working group which developed ISO9126, for his many useful comments and suggestions in connection with this paper.

REFERENCES

1. Cusumano, M. *Japan's Software Factories* Oxford University Press, 1991.
2. ISO/IEC 9126 - Software Product Evaluation; Quality Characteristics and Guidelines For Their Use. Available from the International Organisation for Standardisation, 1, Rue de Varembé, Casa Postale 56, CH-1211 Genève 20, Switzerland.
3. Cusumano, M. 'The Software Factory: A Historical Interpretation' *IEEE Software*, March 1989.
4. Deming, W. *Out of the Crisis* Cambridge University Press, 1986.
5. Kishida, K., et al. 'Quality Assurance Technology in Japan' *IEEE Software*, Sept 1987.
6. Matsubara, T. 'Project Management in Japan' *American Programmer*, June 1991.
7. McQueen, I. *Japan: A Travel Survival Kit, Third Edition* Lonely Planet Publications, 1989.
8. Mikes, G. *The Land of the Rising Yen* Penguin Books, 1970.
9. Sakaiya, T. 'An Ideal Industrial Society: The Two Faces of Japan' *Time International*, August 1991.