Meiji Institute for Global Affairs

MIGA COLUMN GLOBAL DIAGNOSIS

July 22, 2016

Tomokazu Hamaguchi

Fellow, Meiji Institute for Global Affairs, Meiji University

Short Curriculum Vitae) Tomokazu Hamaguchi

Joined Nippon Telephone and telegraph Public Corporation in 1967, after graduating with a degree in engineering from Kyoto University in the same year. Engaged in development of computer systems at NTT. Transferred to NTT Data Corporation along with the subsequent spinoff of the related division.

Served as President of NTT Data (2003 – 2007).

Assumed the current position after service as Chairman of the Japan Information Technology Service Industry Association (JISA).

Serves as Outside Director at East Japan Railway Company and some other companies in addition to Advisor at the JISA.

Japanese computer systems - their differences from the rest of the world

“There are various differences between Japan and the rest of the world in respect of computer systems. Here, I will consider those in the aspects of reliability and package use.”

1. Computer system reliability

Trends in information technology (IT) rapidly change. A few years ago, the key word was SMAC (social, mobile, analytics, and cloud). But now, everyone is talking about the Internet of Things (IOT) and artificial intelligence (AI). SMAC, of course, has not vanished; IOT, for example, would naturally not have arrived without it.

At any rate, in today’s world, all sorts of things are linked through the Internet as the point of contact. In recent years, it often happens that accidents occurring at remote locations in railway and subway systems affect completely different locations, due to mutual direct operation. In information systems, too, along with the spread of IoT, accidents at different sites could conceivably affect your own site.

Because technology advances, incidence of this sort of trouble will presumably be prevented as far as possible. However, I believe that, from now on, Japanese must change their thinking on information system reliability.

I have been involved in the development of information systems that form the social infrastructure. One thing that I am always told by customers is “At any rate, please make one that doesn’t stop working.” I suspect that, for some reason, Japanese want to believe that machines, and especially computers, will not break down. To be sure, computer hardware (the mechanical parts) does not break down very easily and can be backed up. Nevertheless, computers are run by the software, of which there is only one. Therefore, if the software has some bugs, the computer will inevitably stop working. It goes without saying that steps are taken to reduce down time as far as possible or to restrict its extent, but systems are nevertheless put down, depending on the location of bugs and timing. At present, the only way to eliminate bugs is to do tests. With large-scale software, however, the number of things that must be checked in such tests is reportedly on a par with those in the case of space stations. In addition, software is constantly being modified, and it is impossible to test everything with each modification. Therefore, it would be impossible to make a big system which would absolutely not go down.

Japanese information systems, and especially those of the infrastructural type, are said to be extremely reliable even in the worldwide context. But efforts to heighten reliability result in a lengthening of test time and increase in costs. As a result, the systems may not be finished in time to take advantage of business opportunities.

This also holds for recent developments. Information systems are now under construction in Japan’s megabanks at substantial expenditures of time and money. The purpose is to bring them closer to systems free of down time. This is why information systems built in Japan cost more.

What about the situation in other countries? Quite recently, the system at the stock exchange in Singapore went down. As this suggests, a fair number of even large-scale infrastructural systems are put out of operation by failures a few times per year. This does not result in serious disruption, however, because measures are readied in advance for operational suspensions. There may also be a difference of culture, that is, a belief that computers will inevitably break down.

I doubt any people who use computers think that Windows will be completely free of bugs. If Windows personnel sought the same level of reliability as is expected of Japanese systems, the software would probably never be placed into use. Users, too, give precedence to convenience.

Speed is important even today. But in the coming age, an inability to build systems swiftly will make it impossible to cope with the trends of the times. A requirement to exclude any down time will drive up investment costs, prolong the work, and result in failure to address the prevailing wants and needs.

From the first, the rest of the world never demanded perfection of computers. Efforts were instead focused on how to develop them quickly and acquire a position of competitive advantage. A certain level of reliability was, of course, necessary.

But emphasis instead fell on preparation of contingency measures for failures and on shortening the time required for placement back into service.

In the IoT age, I think it is high time for Japan, too, to get over the myth of fail-safe computers.

This does not mean that reliability ought to be ignored. The point is instead to consider a proper balance with many other important factors.

2. Use of packages

There is another respect in which the situation in Japan differs greatly from that in the rest of the world.　It is the rate of use of software packages (PKG).　Here, I also want to consider the so-called joint use (development) of the same software with other companies and use in cloud arrangements (sales force type)　(Naturally, I am speaking about business-use software, excluding operating system (OS) software and other control-use software).

In other countries, the rate of PKG use has been over 50 percent for a couple of decades. In Japan, in contrast, it is probably only around 20 percent even at present.　Software is still something that is developed by human beings, and the development of new types therefore takes some time and money.　The use of existing software including PKG is the cheapest and quickest means of development　(Although machines may possibly become capable of writing software in the future, human beings will probably still have to determine the specifications).

The organizations typically spending a lot of time and money on development of software through isolated made-to-order programs are local governments in Japan. While there are certain differences among local governments owing to size and other factors, I believe their basic work is fundamentally the same.　When I mention this to the heads of local governments, they say that the same software should be capable of use by all. But as one goes down the ladder of subordinate units doing the actual work, they all reject the idea.　The story used to be the same at regional banks, which would never lend an ear to any suggestions of joint use, for example. But after the bursting of Japan’s economic bubble, they made reviews of their systems as part of larger programs for reconstructing their management. As a result, almost all regional banks now practice joint use or PKG use.　The lesson is that organizations can do it if they really want to. This kind of use is definitely something they ought to practice.　In the industrial sector, joint use is more difficult because of the differences between business categories. But lately, PKG use in forms such as SAP, Oracle, and domestic ERP has been spreading. Even so, there are apparently few companies that use PKG without any customization (addition of distinctive features to the PKG).　The coming years will see an increase in the amount of computerization development in connection with IoT and other trends as well as further globalization. In any case, organizations will not be able to compete without a reduction in the load of software development.　Companies should draw a line between competitive and non-competitive areas, and boldly pursue use of PKG and the like in the non-competitive ones.