

Software Gap between Japan and US

Tetsuo Tamai
College of Arts and Sciences
University of Tokyo
tamai@komaba.c.u-tokyo.ac.jp

1 Characteristics of Software Development in Japan

There have been a good amount of discussions that compare software technologies and industries of Japan and the US [1, 2, 3, 4, 6]. Some common observations on the characteristics of software development in Japan may be summarized as follows.

technologies

- Informal problem analysis methods are relatively well practiced, while formal or semi-formal requirements definition methods are not popular [6].
- When new technologies are being introduced, Japanese software engineers do not necessarily require tools that support those technologies, while US engineers seem to assume tool support. This phenomenon can be symbolically illustrated by the fact that the concept of CSCW was first advocated in the US and not in Japan, where people are supposed to be keenly conscious about teamworks and coordination.
- Software reuse is relatively widely practiced.

management

- The time spent in the planning phase of software development is very long; sometimes it extends as long as a half of the total project period [2, 5].
- In the analysis phase, the process of reaching consensus among software engineers, users and managers is more emphasized than the results of analysis.

industry

- Among many characteristics of the software industry in Japan, the most conspicuous is the low percentage of package software that occupies in the total revenue. It is about 10% and has been generally stable in the last decade.

2 Change of Perception on Japan/US Software Gap

All the characteristics above lead to the following results: Japanese software industry is good at producing customized software that meet sophisticated and minute user requirements with relatively high productivity and high qualities. On the other hand, it has not been successful in producing platform software, shrink-lap software and, in general, very innovative, globally popular software.

Around 1990, there was a perception in the US that the competitive power of Japanese software industry was going to be strong and unless appropriate countermeasures were taken, the software industry in the US would follow the path of the steel, home electronic appliance, semi-conductor, car and other industries that had been defeated or severely damaged by Japanese counterparts. Especially, the high quality and high performance realized by “software factories” were perceived as the source of impending threat [3].

In five years, the perception has drastically changed. Today, American software producers are extremely confident about their strength and the dominant share they have in the global market, whereas Japan is seriously worried that the gap in software technology and industry strength between US and Japan is widening.

Coincidentally, the time the Japan-software-threat-theory was dominant in the US was when the bubble economy in Japan was at the end of the peak. The economy faced a severe depression after that and the computer and the software industries were particularly damaged.

In my view, the software industry also had software bubbles in the late 1980's, that is the demand for software apparently increased more than reality by the factors:

1. similar software being constructed user by user;
2. similar software being developed repeatedly within the same organization;
3. orders of software production being subcontracted and sub-subcontracted down the industry hierarchy.

This was the reason the shortage of programmers was considered a serious problem at that time.

Suddenly, the situation changed and the bubbles burst. The apparent advantage of Japanese software technology and industry turned out unfitting to the production of widely influential software and the acquisition of a large share in the global software market.

3 So what?

The US software industry and the software research community seem to have overreacted to the ungrounded knowledge of Japanese software competitiveness. But that should have helped raising governmental funds for software research and industry stimulation. Now, Japanese software engineers, researchers and business managers are too pessimistic. It

may take time to fill the gap but the advantage of user-satisfaction-conscious and quality-conscious development may bring good results, once the transition to product-oriented supply and consumption of software be successful. The current strength in game software may give hints to Japanese software related professionals.

References

- [1] Notkin, D. and Schlichting, R. D.: Computer Science in Japanese Universities, *IEEE Computer*, May 1993, pp. 62–70.
- [2] Belady, L. A.: The Japanese and Software: Is it a good Match?, *IEEE Computer*, June 1986, pp. 57–61.
- [3] Cusumano, M. A.: *Japan's Software Factories*, Oxford University Press, 1991.
- [4] Matsubara, T.: Project Management in Japan, *American Programmer*, Vol. 4, No. 6 (1991), pp. 41–50.
- [5] Tamai, T. and Itou, A.: Requirements and Design Change in Large-Scale Software Development: Analysis from the Viewpoint of Process Backtracking, *Proc. 15th International Conference on Software Engineering*, May 1993, Baltimore, Maryland, U.S.A., pp. 167–176.
- [6] Tamai, T.: Current Practices in Software Processes for System Planning and Requirements Analysis, *Information and Software Technology*, Vol. 35, No. 6/7 (1993), pp. 339–344.
- [7] Tojo, A. ed.: *Report on Japan-US Software Gap*, Japan Electronic Industry Development Association, March 1994, in Japanese.