
William V. Rapp

Working Paper No. 94

Working Paper Series

Center on Japanese Economy and Business

COLUMBIA BUSINESS SCHOOL

William V. Rapp

Working Paper No. 94

William V. Rapp
University of Victoria

Working Paper Series
Center on Japanese Economy and Business
Graduate School of Business
Columbia University
June 1995

William V. Rapp

University of Victoria, Center for Asia Pacific Initiatives, Victoria, B.C., Canada V8W 2Y2. Tel. 604-721-7020 (Fax: 3107)
April 1995 (Revised)

Abstract

The paper analyzes the development and structure of the Japan's software industry, focusing particular attention on the failure of government support policies to nurture an internationally competitive industry and the success of foreign software producers in the Japanese market. The research adopts an evolutionary approach to explain Japan's problems in building a competitive software industry except in games, challenging the validity of standard explanations such as cultural disadvantage and failure to adopt best practice. The pattern of development in the Japanese software industry results from strong user preference for customized software over packaged products. Large users develop highly customized software systems that reflect their unique organizational structures and operating advantages. Software is viewed as an input into the production process, rather than an output to be independently marketed. As such, the structure of the software reflects the administrative heritage of major industrial users as much as developments in computer technology.

The tendency to emphasize customization favors intra-industry software advances over inter-industry expansion. Moreover, high demand for customization combines with large variation in operating systems to fractionalize the industry and frustrate government support policies. Even though the Ministry of International Trade and Industry followed the same industry support model that proved so successful in manufacturing, an independent world class software industry has not developed. The preference for customization has also allowed foreign software producers to successfully pursue a hub and spoke strategy. Foreign producers have been able to expand their user base, thereby lowering unit cost and putting local software developers at a further disadvantage. The research suggests that this situation will continue.
INTRODUCTION

The purpose of this paper is twofold. First, it seeks to explain the development of Japan’s software industry and the failure of government support policies using the evolutionary paradigm. The evolutionary approach proves superior to alternative theories such as quick-adoption-of-best-practice and culturally determined advantage in explaining the Japanese software industry’s failure to become globally competitive, except in games, even though the government followed its traditional and time-tested development model. The second purpose is to then assess the industry’s current strategic structure and to forecast its likely evolution in light of key business and economic variables taken together with the roles, administrative heritages, and motivations of users, suppliers, and government policy makers.

Existing explanations of Japan’s industrial competitiveness give radically different predictions for the likely success of the country’s software industry. The best practice hypothesis states that in industry after industry, managers of leading Japanese firms have proven themselves more capable than their counterparts in Western and other Japanese firms because they can easily adapt to technological and market opportunities. Under this hypothesis, Japanese software firms would be expected to quickly introduce technical and organizational changes necessary to take maximum advantage of changes brought on by new software markets and technologies, such as with network servers and open systems. Indeed, this should occur ahead of their Western and Japanese counterparts. This has not happened.

Another analytical approach to organizational competitiveness is the view that managers’
capabilities are culturally or institutionally bound (Aoki 1990). Under this view, Japanese software companies are at an inherent cultural and linguistic disadvantage and should not be expected to be internationally competitive (Delaney 1994). In reality, Japanese firms are not competitive in most software applications, but they dominate the global market for game software. The cultural disadvantage approach cannot explain this outcome.

The evolutionary paradigm, by contrast, argues that a firm's corporate management is founded on an evolutionary process that reflects the development over time of company-specific sets of strategic rules and routines. These rules and routines govern strategic and operational decisions and implementation as affects R&D, production, human resources, finance and market expansion. As such, they determine organizational success and failure. The evolutionary theory predicts that even leading Japanese firms can have difficulty adapting to new global competitive conditions; however, where conditions replicate other Japanese successes, Japanese software can compete. This has occurred.

The Japanese software industry is characterized by strong user preference for customized software over packaged products. Large users develop highly customized software systems that reflect their unique organizational structures and operating advantages. As such, the structure of the software reflects the administrative heritage of major industrial users as much as developments in computer technology. Software is viewed as an input into the production process, rather than an output to be independently marketed. Since software is a production input, sales do not show up as unit sales of packaged software; rather, software sales are one component of finished product cost (Dalton and Genther 1991).

The evidence supporting the evolutionary hypothesis allows predictions about which
software is favored by Japan's competitive strengths. Those strengths favor graphics, games, and certain multimedia applications as well as the unique customized systems that support Japan's competitive prowess in steel, autos and consumer electronics. The tendency to emphasize customization favors intra-industry software advances over inter-industry expansion. Moreover, high demand for customization combines with large variation in operating systems to fractionize the industry and frustrate government support policies.

This conclusion is consistent with the findings of Teece, Pisano, and Shuen (1990) on corporate strategy that combines evaluation of firm resources and capabilities with competitor analysis in a dynamic context. It also draws on the conceptual tradition of Schumpeter (1947) with respect to the importance of firm innovation and creative destruction, as found more recently in works by such authors as Abernathy and Utterback (1978) and Tushman and Anderson (1986). At the same time, it relates directly to the growing body of economic analysis that argues for the importance of what goes on within and between organizations (e.g. Coase 1937; Cyert and March 1963; and Ostry 1990), and it supports research indicating corporate history and administrative heritage matter in corporate behavior and successful competition (e.g. Bartlett and Ghoshal 1989; Nelson 1991).

Industries in which Japan has shown particular competitive success are key in this regard. They are the most visible aspect of Japanese global competition and are thus the source of considerable software demand. Steel, ships, autos, electronics, machine tools and their related industries represent the cutting edge of Japanese competitiveness worldwide as noted in Abegglen and Rapp (1972), Aoki (1988), Anchordoguy (1988, 1990), Clark and Fujimoto (1991), Cusumano (1985, 1988, and with Rosenbloom [1987]), or Rapp (1992). Japanese firms in these
industries do particularly well in large complex manufacturing and assembly industries involving close coordination of several suppliers and different parts of the firm. Customized software contributes to this kind of production integration. Software is customer-driven and becomes an integral part of a firm’s global market advantage.

Further, user industries are often benefit from controls that generate economic rents, providing a pool of funds to develop productivity-enhancing software. Economic rents are used to upgrade products and technology via more sophisticated software, setting the stage for the next wave of high-quality low cost products. So far it has been primarily foreign software that has been adapted and customized by manufacturers, a trend that is likely to persist.

The Ministry of International Trade and Industry (MITI) is trying to shift the nature of government support for Japanese software developers (excluding game developers), hoping to promote small and medium size companies whose software seems to be in demand. However, the ministry still seems to be treating software as an output rather than primarily as an input into other industries. In this sense, there is little change from the MITI strategy that failed in the late 60s and early 70s. A better strategy would be to go with the economic and competitive flow and promote industry enhancing software that will assist major Japanese users to maintain their global competitiveness. Otherwise, the government’s options appear quite limited.

Major users are conservative in their use of software, changing their systems only incrementally after thorough testing. They will not introduce a new system just because it is Japanese produced. The risk is too great that a glitch in a complex integrated software system will disrupting their basic business, and there is no perceived benefit. This concern with the integrity of the business system, combined with economics, provides an inherent advantage to U.S.
software producers who are localizing and adapting an existing piece of software over Japanese producers who are trying to win over Japanese corporations to new products. In turn, because large users focus on localization, adaptation, and customization, the Japanese software industry does not use its available programming and software engineering resources to develop packaged software to sell into the global market, except for games.

The globalization of Japanese corporations and their need for global systems integration has not affected demand for customized software. While foreign staff may insist on the purchase of certain prepackaged software to work in their market, Japan is unaffected, neither controlling nor being influenced by the decision. The company system is not integrated globally, and among those interviewed for this research, there appear to be few plans to move in that direction. Further, Japanese staff are trained primarily in the employer’s computer language, limiting familiarity with a range of international computer languages and hardware platforms. Also, many Japanese EDP managers are not computer specialists; instead, they are corporate managers who believe that the firm’s software development should stay focused on maintaining competitiveness. This tendency means Japanese software developers are primarily organized along industry lines, a patterns that generally applies to independent developers as well (Coulitas 1994).

There is sometimes cross-shareholding between developers and major customers. If the affiliation is fairly close, it can also involve an exchange of personnel. But with the exception of banks, software system usage does not seem to extend to major users’ keiretsu suppliers. That is, keiretsu firms are not linked through software in a common integrated system. However, supplier firms will usually have their programs developed by the keiretsu’s software development company. As much as 50% of the sales of a major user’s software supplier may be to companies
in the same group as the user, including those in which the user owns shares.

Vertical integration and group sales may not be a great benefit to the developer though, since they do not spread the fixed development costs over a wide enough user base. This situation merely serves to continue the fragmentation of the Japanese software market. At the same time, foreign suppliers gain access to most of the market by working with each of the industry software houses and major systems suppliers— including Hitachi, NEC, Fujitsu, IBM, and DEC. Foreign suppliers localize their software to a Japanese system supplier’s system and platform, then they let the system houses or users handle the customization to users’ specific requirements. In this way, foreign software suppliers develop a series of plug compatible software modules that fit into the users’ existing systems, without threatening either a system’s overall viability or major system producers’ relationships with users.

This suggests that the better a foreign firm is at working with systems suppliers to localize and adapt its application programs, the more successful it will be in the Japanese market. This is because most of the Japanese software market, even for personal computers, remains corporate. Households are still primarily using Japanese word processors instead of PCs, though this may change as low cost U.S. PCs using localized Japanese software capture more of the home market (AEA 1992). The entry of Dell and Compaq longer term is therefore clearly beneficial for Microsoft and Lotus, while the popularity of U.S. software products helps Dell and Compaq.

Still, for the time being the main market is corporate and because the industrial distribution system (unlike the retail distribution system) is compact and efficient, the foreign software developer is not at a competitive disadvantage compared to local suppliers. The market potential thus appears good while ease of entry is improving. The relative openness of the market as
compared to semiconductors has also proven helpful. Indeed, there is a marked contrast between the protective tools available to the government at the time they targeted semiconductors for development and what they face in software today. At the same time, one reason the computer market remains highly fragmented is MITI's early computer licensing policies which gave each manufacturer a different foreign partner. When those partners were forced to exit, Japanese licensees and their customers were technically isolated. They also had to adapt to software developments abroad on their own since their normal source of innovation had evaporated.

As long as software continues to advance rapidly, U.S. producers seem to be at an inherent advantage. They can maintain this technical edge more easily in software than hardware because development is less capital intensive and the primary input, skilled people, is fully tax deductible. Further, as noted above, Japanese customers have very integrated systems and are quite risk averse with respect to the software they use. Thus they avoid experimentation and instead emphasize software that is totally debugged and reliable (AEA 1992). There is little benefit in pushing the development frontiers and then risk having your production system crash.

The paper concludes that presently fragmented operating systems and software application patterns will persist. Standardization is unlikely, except for some work stations. Further, since software is a labor intensive industry and Japan is a high-wage country with a shortage of system engineers, development of non-customized software in Japan is relatively expensive and, with the strong yen, internationally uncompetitive. Indeed, the cost of localization and customization keeps even converted foreign software expensive, though less so than locally developed products. User preference for customization robs Japanese developers of the large sales base necessary to lower unit cost. Unless the preference for customization changes, internationally competitive,
locally developed packaged software is unlikely to develop outside of the game sector. At the same time, there is no economic or management incentive to end customization, given a firm’s huge sunk costs and strong desire to support its unique competitive operating advantages in its basic businesses, which are not software. This means only games, niche software such as Japanese language word processing programs, and software incorporated in competitive manufacturing systems like steel or automobiles will be competitive.

The difficulties of the Japanese software industry have caused some to wonder if Japan can successfully enter the information age (Coulter 1994). The concern is that the new techno-economic paradigm described by Freeman and Perez (1986) will require a stronger software industry than seems possible in Japan without significant institutional and cultural changes. However, this paper shows that such concerns emphasize inter-industry developments too heavily, slighting intra-industry development. They also focus excessively on trends in the U.S. It is important to recognize that major Japanese companies are developing and using very advanced software to maintain or improve their global competitiveness in key industries. Their customized or in-house development approach may represent an alternative way to enter the information age.

**EVOLUTIONARY APPROACH**

The development of Japan’s software industry offers an opportunity to test Nelson and Winter’s evolutionary theory of the firm (1982) as an explanation of the industry’s weak competitiveness. The research also indicates that some of the intra-organizational concepts the theory presents can be used to analyze relations between institutions as well, such as those between MITI and large Japanese firms that are major software users or producers. The analysis
is then used to explain some important issues regarding Japan's software industry: the variations in the performance of companies producing games compared to other computer software, the difference in outlook in Japan and the U.S. for substitution of networks for mainframes, and the continuation of large Japanese users' preference for customized rather than packaged software. The analysis emphasizes the differences in user and producer costs, which have influenced the industry's structure and the government's failure to establish a policy that would lead to the creation of a Japanese Novelle, Oracle or Microsoft.

In their seminal work on the evolutionary theory of the firm, Nelson and Winter (1982) postulated that a firm's corporate management is founded on an evolutionary process reflecting the development over time of company-specific sets of strategic rules and routines. This intra-firm view has been extended to explain Japanese corporate success in studies such as Baba (1989) and Rapp (1992 and 1993). Under the theory, adoption of radical changes in organization and technology is difficult even if one recognizes that the failure to change is not the most efficient or the best available practice. This is because any proposed change must accommodate the existing organization and past practice. This hypothesis is thus consistent with studies emphasizing the importance of administrative heritage or legacy systems on corporate decision making, including software selection (Baba, Yasunori, Takai and Mizuta 1993 and Steinmueller 1993).

Nelson and Winter argue that Japanese competitive success results from logical adaptations to business, economic, and political circumstances over many years. Successful forms, policies and procedures developed over time turned out to be extremely competitive in current world markets, at least for certain industries and products. This hypothesis assigns less universal knowledge, foresightedness, or instantaneous best practice adoption to Japanese corporate
managers than is often assumed by other analysts, including many neoclassical economists. Though the competitive strength is still real, it is not necessarily all-encompassing or clearly transferable to different products or competitive environments (Florida and Kenney 1991).

The Nelson-Winter analysis differs from studies by analysts like Womack, Jones, and Roos (1990), Krafcik (1988), or Dertouzos, Lester, and Solow (1989) who argue that the competitive advantage of Japanese firms lies in lean production systems which can be applied to any environment or industry. The excellent study by Cusumano on Japanese software factories (1991) implies that U.S. software producers may face a challenge from Japan as lean Japanese production systems substantially reduce the time and cost associated with product design and production. The logical conclusion is that U.S. corporations should—and can—copy this system, which is best practice, to become equally efficient and cost competitive in a given industry, including software.

Another analytical approach to organizational competitiveness is the view that for cultural or institutional reasons, managers in leading Japanese firms are more capable than U.S., European and Japanese counterparts. Under this hypothesis, a given corporate organization and technology are largely determined by environment, in particular the institutional structure in which a firm operates and its changing relative supplies of capital, labor, and land. A large population with limited arable land, for example, results in relatively expensive real estate, forcing adoption of space saving technologies, including inventory reduction and just-in-time ordering systems. Under this scenario, the Japanese environment is the source of competitive strength. Given its disciplined, well-educated labor force, strong corporate and government support, and extensive financial and computer resources, Japan should easily become a major competitor in software, provided it makes a determined strategic effort of the sort it made in other technology-intensive
industries. In reality, despite success in computer hardware and semiconductors and numerous
government initiatives targeting the software industry going back to the 1960s (Anchordoguy
1989), Japan has largely failed to developed a competitive domestic software industry—except in
the unsupported area of games (Rapp 1993A).

The reasons for this situation are far from clear. Some American policy makers and
corporate executives claim credit for implementing an astute global strategy that facilitates more
rapid adoption of best practice by U.S. firms while protecting producers' intellectual property
rights and market access. This view is similar to the best practice hypothesis. Others have noted
superior U.S. cultural attributes with respect to software development, favoring U.S. program
developers over their Japanese counterparts (Delaney 1994). Recent developments in Japan’s
software industry thus offer an interesting opportunity to test the relative merits of three leading
analytical approaches in explaining Japanese competitive development in software.

It is the Japanese computer industry’s evolution which in fact has been the primary
constraint on users’ ability to adopt best practice, even such superior developments as
substitution of client (network) servers and workstations for mainframes. This in turn has
constrained Japanese suppliers from diverting programming resources to these new technologies.
The major integrated suppliers maintained a strategy of separate mainframe platforms to hold their
customers more closely. They combine this approach with improved vertical compatibility
between the mainframe and their client servers, work stations and PCS by using families of
microprocessors (MPUs). These processors, though compatible with other operating systems, are
optimized for the separate platform. This approach has continued industry fragmentation and
made it more efficient for major computer manufacturers and large users to adapt existing foreign
software than to try to develop totally new software on their own.

In this manner, evolutionary analysis helps explain why predictions of U.S. competitive
decline in software (Feigenbaum 1983, Cusumano 1991) have not materialized even as
competition in high technology electronic hardware remains intense.

SOFTWARE INDUSTRY ECONOMICS

Software industry economics differs from manufacturing in ways that effect the motivation
and behavior of producers and users. The bulk of a software program’s cost is in the
development rather than the physical manufacture. Actual production and distribution is cheap
and may be getting cheaper with the introduction of CD-ROMs. Unlike hardware industries,
therefore, in software incremental improvements in manufacturing and process have little
relevance to competitiveness. Instead, the key competitive variables are keeping development
costs down and getting broad global distribution.

The size of the user base, not the cost of production, drives a software developer’s cost
structure. A larger number of users can more than compensate on a per unit cost basis for higher
development costs. This fact combined with the increased utility of a large user base means there
is a strong tendency towards oligopoly or even monopoly in each software market segment (e.g.
word processing, spread sheet, netware, operating system, tax or graphics program), leading to an
even larger user base and lower prices, with stable market shares once growth slows.

This economic structure means the cost per unit for a successful program drops rapidly,
establishing large cost and user-experience barriers that competitors must overcome. Prices will
trend towards equilibrium where the cost of adding a user (reproducing the program) equals the
increased utility to the user. The latter will become relatively constant at a low price once a large user base exists. These dynamics create substantial incentives for software companies to merge and for large developers to have programs that manage and handle the systems of their largest competitors to capture the externalities of increased user interchange (e.g. Microsoft and Novelle). Both these trends further promote the industry's oligopolistic structure.

After initial development, most software program improvements are evolutionary rather than revolutionary. This facilitates the "upgrade" marketing approach to the existing user base with low incremental development costs per unit. It also promotes globalization based on localization and adaptation of existing programs as the expanded user base lowers the average unit cost on both a local and global basis. Most U.S. packaged software companies have developed source codes that are compatible with foreign language translation to facilitate this process.

Copyright is an excellent way to protect such software on a long term basis as protection in global markets is automatic and for the life of the corporation plus fifty years. It is difficult to change the expression of the programming idea since this changes the external benefits to the user and means the infringer does not have access to the user base that drives firm economics. Software piracy remains rampant overseas, hurting developers' potential cash flow; however, for the developer, piracy has the more sanguine effect of undermining local competitors. Also, users gained through piracy may become customers through better intellectual property enforcement.

The unique features of the industry have interacted with different historical legacies and competitive environments in Japan and the U.S. to place the Japanese and American software industries on different evolutionary paths that seem likely to persist. The favorable economics of the software industry do not exist in Japan (except for foreign software developers, games
producers, and one or two packaged software producers) because large users develop highly
customized systems. This limits the public-good character of software in Japan. In the U.S.,
software shares many aspects of a public good. Use by one person does not diminish its use by
someone else (H. See 1992 and S. La Croix 1992). In addition, the utility and value of much
software is enhanced by increased usage in the U.S. More users means greater ability to exchange
programs, to interface through a network using standard protocols and a common programming
language, and to hire workers already familiar with the software.

In Japan, since there is little employee migration between large firms, companies feel little
need for systems compatibility; indeed, shared software usage would actually reveal sensitive
competitive information. Unlike in the US, then, software does not reflect aspects of a public
good, and there are fewer economic pressures towards merger, integration or consolidation.
Rapid per unit cost reduction is not the norm in Japan, but rather the development of a few related
programs applicable within an industry or company group. This creates pressures to reduce cost
via reusable program segments instead (Cusumano 1991).

Offsetting high unit software costs, though, are the competitive advantages conveyed by a
firm’s proprietary innovative processes, the basis on which Japanese firms generally compete with
competitors more likely to emphasize product innovation (Womack, Jones, and Roos 1990 and
Rapp 1992). Under these circumstances, there is little movement in market share among the
leading integrated computer systems producers for mainframe customers (Rapp 1993A).

In games, by contrast, market share can shift as improved technologies are introduced
through aggressive pricing of new machines. There is little cost difference in buying one
company’s game cartridges over another’s and in either case new machines do not play the old
cartridges. Thus, unlike large Japanese software users, compatibility and substitution are not competitive barriers for games producers.

Furthermore, in the game segment, the software is the end product, not a ton of steel or an automobile. As such, there is no organizational risk in switching to a new system while the format and play between systems from a users' standpoint appears relatively negligible. In this type of medium-technology software market for a relatively standard end product, classic Japanese production and global market share strategies seem to work well compared to U.S. producers like Atari. Japanese competitiveness in consumer electronics and an affluent domestic market are also pluses. But when the software represents an input into a production process, the process dynamic and the outcome are different.

This raises the question whether Japanese software suppliers to the steel and automobile industries should be considered part of the manufacturing industry or part of software. We take the position that they are part of the user industry, not software. These suppliers are usually owned by the users and staffed by user executives. They do little outside work and experience little labor migration to other software developers; moreover, they achieve commonality-of-usage efficiencies through industry or group specialization, rather than through broad horizontal growth in user base. Finally, they are dependent on the economic health of the user industry for their own survival. As a matter of fact, results of an industry survey conducted for this project indicate that most software developers working for large users are so driven by the users' organization and demands that there may be no Japanese software industry per se, except for games.

Further, because Japanese firms are not the global leaders in most software, U.S. firms control the migration of competitive advantage in programming and usage to countries like India.
and Russia, where they are making use of low cost highly trained talent. In fact, Japan is now a high cost country for skilled technical labor. Therefore, to become competitive it needs to make software development more capital or machine intensive. Since this goal has proved elusive so far, educational and firm infrastructure exert unusually strong influence on software development. The market has not promoted development of “advanced factors” of production such as software engineers and programming specialists in the type of agglomerative structure Porter says is necessary to achieve and maintain competitiveness (1990).

INDUSTRY AND POLICY-SPECIFIC CHARACTERISTICS

Another factor favoring U.S. developers and continued fragmentation of Japan’s software industry is the strong relationship between particular software and the microprocessors (MPUs) or logic devices that drive computers. Specific MPUs are developed to work with certain types of software, and software programmers develop software for the next generation of MPUs. There are powerful forces integrating MPU producers, programmers, and users to perpetuate the standards serving a given market segment through several generations of technical development. Customers do not want to abandon their software investment just because they buy new hardware, so new MPUs must run old software as well as new. Similarly, software developers want to use their old code to upgrade operating systems and application programs as this increases productivity, improves quality and reduces development time.

Software developers will concentrate on the most popular MPUs, which due to their high development costs are subject to the same user base economics as packaged software. This has lead to an interactive oligopolistic cycle that has combined with the growth in personal computers,
networking, and parallel processing to make MPUs and their related software the defining considerations affecting competition in specific software markets such as PCs (Intel and Motorola), workstations (Sun and Hewlett-Packard) and mainframes (IBM, DEC, Intel, and Motorola). This trend does not look like it will change; indeed, it seems to be getting stronger. That is, while MAC and IBM/PC based chips and software seem to dominate PCs, UNIX based systems are preeminent in workstations, currently the fastest growing Japanese market segment. IBM compatible systems, though, still lead in mainframes.

With these multiple linkages, there is no single MPU/software interaction upon which Japanese firms can focus. Even among the various generic operating systems, each firm has its own unique mode. While this has the advantage for supplier of restricting customer migration, it discourages the development of positive externalities benefiting the whole software industry such as the development of a dominant MPU/software nexus.

In addition, U.S. firms, having learned about the importance of Japanese market share from their experience in steel, autos and DRAMs, entered the Japanese market for MPUs and software earlier. Indeed, Japanese firms' emphasis on DRAM production served to increase the demand for U.S. MPUs, a trend supported by the US-Japan Semiconductor Accord since Japan committed to buying a certain number of U.S. made chips. Because Japanese companies did not yet produce MPUs in quantity, it was logical to purchase U.S. products. This pulled U.S. suppliers into the market. In turn, Japanese firms underestimated the strategic importance of designing and producing MPUs. They mistakenly assumed that, as had occurred other industries, their manufacturing technology would be better and that they would be able to move into logic through licensing once they had achieved dominance as the low cost chip producers via DRAMs.
Shifts in computer technology and market growth towards parallel processing and PCs combined with multiple generations of MPUs (e.g. 16 to 32 to 64) to change the standard dynamic. The development of the silicon factory and customized chip producers also moved production economics against Japanese assumptions. Profits turned out to be in logic and design rather than efficient standardized production, and the shift from producing memory to designing and producing MPUs turned out not to be easy. The very regularity of the DRAM meant design capability was not emphasized by Japanese firms.

The fifth generation computer project tried to replicate the VSLI project with respect to software. But the only large breakthrough was designing a computer capable of handling inference programs, i.e. a hardware not a software development. Further, productivity improvements in software production, like object-oriented software, have been primarily software developments. Machines capable of writing software have not been produced.

In addition, as noted above, an important part of software development is compatibility with existing programs and operating systems since users do not want to relearn how to use particular programs every time they purchase a new machine. Under these circumstances, the most important competitive factor is again one's installed customer base and existing program design or library, not the ability to produce more error free line code in a given time period as initially hypothesized by Cusumano (1991). This creates an unbenefficial cycle substantially different from the one seen in DRAMs.

The cost of line code per user is more important than the cost per programmer. If a given amount of line code is spread over a larger number of customers, the cost per customer falls. When two line codes are incompatible and are directly tied to their respective installed customer
bases, the firm with the larger installed base can sell at a lower unit price, all other things being equal. Large up front development costs, low physical production and distribution costs, and customer inertia all favor this kind of competitive advantage. This explains why many packaged software firms rush new programs to market at very low prices to establish a large user base. As noted above, it also makes acquisition of a firm with an existing program and user base very attractive as an entry or market share expansion strategy for another firm with greater distribution and financial power. Several recent U.S. software acquisitions confirm this.

If Japanese firms were interested in responding to these competitive conditions, they would have several traditional options. They could wait for U.S. growth to slow and for U.S. firms to establish a standard dominant design they could easily emulate. However, there are some difficulties with this strategy. Software control through copyright is for 55 years or more, so Japanese firms can't wait. In addition, it is not clear that any dominant design that works across all market segments will be established. Finally, Japan itself is a relatively large and mature software market: by the time U.S. growth slows it will be slowing in Japan too.

These realities have combined with continued customization and fragmentation to produce some different approaches. There have been significant increases in strategic joint venturing. While a joint venture limits a Japanese firm's ability to capture new markets, it helps the firm retain its domestic customer base. Related to this are government attempts to make it easier for Japanese firms to use foreign developers' existing software libraries. However, decompilation, which allows firms to re-engineer programs to the original zeroes and ones and then to rebuild them into usable programs has evoked a strong U.S. policy reaction.

Decompilation does little to change the user base advantage of the foreign developer. It
also represents just another form of localization that siphons off programming resources from development to copying to localizing the copied product. Further, this type of re-engineering results in "legalized dumping" of localized U.S. software onto the Japanese market. De facto, it continues U.S. dominance and hurts the evolution of an independent Japanese software development capability. It also makes Japanese firms dependent on a continuous stream of such reengineering for upgrades and new products. This accentuates the existing trend where hardware producers have organized themselves to assist foreign independent software developers in localizing their software so a shortage of attractive software does not threaten the hardware client base. This only gives further impetus to the trend of Japan's programming resources flowing towards localization and adaptation than towards development.

With the exception of Sega and Nintendo in games, there has emerged no dominant Japanese firm with a global software presence. Further, each large integrated systems producer and its major users have the resources to support their own unique local programming and software packages. Conversely, agreeing on a common design or standard creates strategic problems for system producers as it opens their "captive" customer base to poaching by other Japanese firms. This has been a particular dilemma for NEC in the PC market but applies more generally as well. This again tends to keep the market fragmented, frustrating the development of a dominant software design despite MITI's efforts to do this for UNIX via the Sigma project.

THE PARADIGMS

How did this divergent industry organization and economic structure emerge? Its current form evolved historically from specific computer and computer software policies. In addition, the
structure and practices of the industry have been influenced by factors such as Japan's economic structure, the labor force, employment practices, government-business relations, management styles, the financial system, international trade, and investment policies. The importance of each factor to software varies with the particular competitive context, the type of software and its market. That is, mainframe software and its competitive dynamics are related to but different from software for workstations and networks, while games represent another variation altogether. Each situation must therefore be analyzed for its critical elements.

Research results indicate Japan's current industry structure comprises the intersection of two major trends. The first is the technical development of the computer industry and the second is the increasing sophistication and administrative heritage of major users. Any explanation of these strategic developments must address and reconcile the persistence of customized software as Japan's dominant software market, despite high costs and MITI policies promoting standardization, and Japan's global success in standardized game software. An explanatory hypothesis must also explain the existence of multiple platforms, the strong support for localization and adaptation of foreign software, the push for allowing decompilation, the shifts in user preferences toward flexibility and open systems combined with customization, and users' increased desire for compatibility between their software needs and their basic management goals. Only an evolutionary approach addresses all these issues.

The key competitive factors affecting large firms seem to be their administrative heritage, their process and systems integration, and constant foreign product innovation and development. First, due to multiple licensing arrangements orchestrated by MITI in the early 1960s, each major computer system company has a different historical antecedent for its operating system.
(Hitachi/RCA then IBM, NEC/ Honeywell, Oki/Sperry Rand, Toshiba/GE, Mitsubishi/TRW, and Fujitsu/its own then IBM). In turn, to keep their customers dependent on them for support, upgrades, and application development, suppliers did little to encourage interoperability between the disparate systems (Anchordoguy 1989; Baba et al 1993; Cottrell 1993; Rapp 1993A; and Coultras 1994). In addition, government subsidies for mainframe computer development and sales encouraged suppliers to offer their customers highly customized software packages using their proprietary operating systems as a further subsidy and lock-in device. While IBM was required by a 1960s U.S. anti-trust decree to unbundle its software and hardware sales on a global basis, Japanese firms were not.

As Marie Anchordoguy (1989) quotes one informed observer, “In the 1960s, Japanese companies almost gave us their computers. They wanted us to work with them and give them feedback to improve their systems. If we wanted new software, they would give it to us for free. They were losing money, but they looked at it over the long-term.” Japanese firms' competitive compulsion to defend existing customers and markets (Ohmae 1991; Rapp 1994) further hardened this situation and encouraged computer manufacturers to try to extend this strategy to network servers, workstations, and PCs. A typical case is NECs reluctance until recently to adopt a fully compatible MSD-DOS PC operating system. Thus, historical ties have left Japan with an installed computer base of multiple systems and incompatible platforms, particularly for mainframes, which are strategically difficult for users to change. This conclusion is strongly supported by responses to a questionnaire distributed to large users, integrated suppliers, and independent software developers, and is further supported by interview evidence.

The reason it is now so difficult for users to change their systems or approach to software
development is that across a wide range of industries over the last twenty to thirty years, various users have written tens of millions or even billions of lines of customized computer code for their existing operating systems. It is too expensive and time consuming to convert this software, which is mostly written using COBOL, to work with newer programming languages such as C or C++. For example, at a rate of six to ten million lines a year, it could take some firms interviewed thirty to a hundred years to convert their existing programs to work with the newer operating systems. It would also divert large numbers of scarce systems engineers from the more important task of developing programs for new business requirements or of maintaining existing ones. Thus, since the existing programs still accomplish basic business goals, large users see little reason to spend the time, effort, and money to convert them to the newer systems even though there may be some loss of efficiency, operating flexibility and inter-operability.

There are several other reasons for the slow pace of user conversion. First, there are no programs to convert large COBOL based systems to C language. Apparently, "the Americans have not developed one." Large packaged software sales in the U.S. made it pay for the packaged software developers to convert their programs; however, due to the emphasis on customization in Japan, the packaged market is small (table below), and no one user is large enough to justify developing such a program for its own use.

Also, because most programming has been done in COBOL and employees do not frequently change jobs (Pucik 1984; Abegglen and Stalk 1985; Rapp 1992), there are few programmers trained in C language. Meanwhile, firms would rather use the programmers they have to develop programs for new requirements, i.e. at the margin of their systems. Even if the code were rewritten, considerable testing would be required to make sure it worked. During this
testing period the old system would have to run in parallel requiring additional time, skill and 
knowledge to maintain it.

Finally, the existing systems, while perhaps not as flexible, are fully expensed. So there is 
no compelling business reason to totally and rapidly shift to the newer systems. Indeed, in many 
industry associations there is a technical group meeting regularly to discuss the latest software and 
the best way to introduce it into industry practice. Everyone recognizes that the rapid 
introduction of the new client server based technologies would be expensive for all and would not 
create any real competitive benefit since all would follow. The best approach in the current 
adverse economic climate becomes for everyone to shift at the margin.

This marginal-shift approach has occurred despite government efforts to promote a switch 
to newer software technologies and the development of a strong packaged software industry. 
After several false starts in the 1960s and 70s (Anchordoguy 1989), MITI made a strong effort in 
the early eighties to promote Japan’s development as a leading edge global software supplier, 
including promoting such areas as artificial intelligence and object oriented programming. 
Seemingly successful government policies in computers and ICs were applied to initiatives for 
software (Rapp 1993 A) such as the ICOT, Sigma and TRON initiatives.

Under these programs, MITI used an evolutionary approach that promoted software via 
the ‘fast follower’ concept they had used so successfully in industries like steel, autos, computers 
and semiconductors. In this model, competitive advantage emerges dynamically from managing 
the international product cycle. Foreign entry is controlled through licensing and technology 
transfer arrangements that enable Japanese followers to absorb foreign technology then improve it 
and lower production costs through attention to detail and organization, finally becoming globally
competitive. Capturing and building on a rapidly growing domestic market for a standard product is usually a key consideration for success.

This is the paradigm of "quick adoption of best practice", and if it had worked, this policy would have demonstrated the superiority of leading Japanese firms' management techniques in yet another industry as predicted by analysts such as Feigenbaum (1983) and Cusumano (1991, 1992). The prospect (or fear) of such an outcome captured the imagination of many U.S. and European policy makers and executives (Anchordoguy 1989). Yet, interviews with MITI officials and industry analysts indicate that while the goal of developing a globally advanced and competitive software industry has not changed, there is growing recognition that initiatives to date have been weak and the old "fast follower" model will not work for software. Coulitas (1994) reaches the same conclusion based on similar interviews.

MITI is therefore searching for new, more successful policy approaches to encourage software industry development. What is not clear is whether a new policy model will address the realities of the industry's evolution or the reasons the old model failed to achieve its usual results. This is because policy makers have industry development rules and routines that have been very successful, and it is organizationally difficult for them to change them (Nelson and Winter 1982). Indicative of this is that while MITI is trying to promote greater use of computers and the information highway, the department in charge of the promotion effort is just now connecting to the Internet and senior officials have little experience using computers.

Further, it is clear from Anchordoguy's research (1989) that an important ingredient in promoting computer hardware sales in the 1960s was the government's ability to control a user's mainframe purchase decision through the foreign exchange law and import licenses. Technology
import licenses and administrative guidance were used to manage IBM, for example. These measures no longer exist because policy success in developing strong internationally competitive corporations has led to a massive export surplus, repeal of the foreign exchange law, and heavy U.S. government policy involvement. MITI can no longer pressure users to buy Japanese software or switch systems contrary to their interests. The U.S. government has also taken a more proactive stance in objecting to actions such as mandating the use of TRON in Japanese schools (Coulitas 1994) or changing intellectual property laws protecting software (Rapp 1986).

Furthermore, success in computer hardware, semiconductors, and consumer electronics has limited the availability of software engineers and programmers. Individuals with the necessary technical background to become software experts have been employed and trained to develop hardware rather than software. Similarly, large users like Nippon Steel and Toyota can recruit the software engineers they need to maintain and develop their proprietary systems. Few are left over to support a large independent packaged software industry. Therefore, in software Porter's (1990) positive agglomeration effects do not exist.

In addition, the major integrated suppliers producing operating and application software to run on their proprietary hardware share a common historical experience with managers of other internationally successful firms. Such companies have historically captured the growth in Japan to build experience and market share (Abegglen and Rapp 1972, Abegglen and Stalk 1985 and Rapp 1992). Since the Japanese market for information services is growing, suppliers must capture the world market growth represented by Japan to become competitive and be successful fast followers in software, just as they did with government help in the 1960s and 70s in computers (Anchordoguy 1988, 1989) and in the 1970s and 1980s in semiconductors (HBS 1989; Rapp
Yet, this is not happening in software.

In fact, the integrated suppliers are actively supporting the localization and adaptation of foreign application software to their systems (AEA 1992, Rapp 1993A), increasing U.S. firms’ user base, lowering their per unit costs, and improving their global market share. Neither are integrated system suppliers reallocating scarce technical development personnel from hardware to software. Their strategic emphasis remains hardware. Similarly, major users have established captive software subsidiaries to better manage the development of proprietary software for the parent and affiliates and to improve career opportunities for their software personnel.

Why are they doing this, when the “easy adoption of best practice” model would argue they should appropriate and build on such advanced software for their own purposes and the “cultural” model might argue that superior Japanese management should organize itself to replicate it cheaper and better? Evolutionary theory appears to offer the appropriate insights in that each market segment has evolved differently and segments are not growing equally. Looking at the table below, mainframe software developed first. As a large but mature market, mainframe sales and software are growing much more slowly than minis, work stations, and PCs and their software. Further, mainframes are used more for distributed processing, data base management and overall Electronic Data Processing (EDP) than for specific task-oriented activities.
Growth Rates for Software 1990-91

<table>
<thead>
<tr>
<th>Product</th>
<th>Custom</th>
<th>Packaged</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mainframe software</td>
<td>5.5%</td>
<td>-8.8</td>
</tr>
<tr>
<td>Mini Computer software</td>
<td>22.1</td>
<td>36.3</td>
</tr>
<tr>
<td>Work Station software</td>
<td>33.9</td>
<td>18.2</td>
</tr>
<tr>
<td>PC software</td>
<td>18.9</td>
<td>3.7</td>
</tr>
</tbody>
</table>

Yet, the mainframe software market is large. Yen 360 billion in 1991, of which Yen 352 billion is customized, with growth for 1991-94 expected at 5.0% per year. Thus, customized mainframe software sales will be stable while growth opportunities are in downsizing and applications, both custom and packaged. In addition, the customized mainframe software market is larger than all other software markets combined and is likely to remain so for awhile, absorbing considerable financial and personnel resources. This view is confirmed by the interviews with large users who generally agreed it would be five years before they converted the various tasks that they envision changing to network servers, work stations and PCs. Even then, users forecast that 30% to 50% of total EDP costs would remain mainframe-related and the associated software would continue to be customized.

A 1992 MITI survey (JISA 1992) produced results consistent with the findings of this project. Even when users switch to network servers, work stations and PCs, customization needs do not change appreciably because users want the new software to be integrated into the company’s overall system. Users indicate that except for word-processing and spreadsheet programs even the packaged software they bought had to be adapted to their systems. This is true despite customization expenses that can run ten times the cost of similar packaged software.
This could increase the overall cost of a program 200% in that customization would equal around 70% of the final cost of installing the package.

From this data it is clear that although highly visible, packaged PC software is a small part of the Japanese software market—about 3%—and likely to remain so for the foreseeable future. Customized mainframe software will continue to represent the bulk of the market.

### Japanese Software Market (Yen billions and Growth in %)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Customized</td>
<td>Y 562 b</td>
<td>12.2%</td>
<td>7.1%</td>
</tr>
<tr>
<td>Market</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total Packaged</td>
<td>29</td>
<td>3.2</td>
<td>13.8</td>
</tr>
<tr>
<td>Total</td>
<td>591</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mainframe Custom</td>
<td>352</td>
<td>5.5</td>
<td>5.0</td>
</tr>
<tr>
<td>Mainframe Packaged</td>
<td>8</td>
<td>-8.0</td>
<td>7.2</td>
</tr>
<tr>
<td>Subtotal Mainframe</td>
<td>360</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mini Custom</td>
<td>85</td>
<td>22.1</td>
<td>7.3</td>
</tr>
<tr>
<td>Mini Packaged</td>
<td>2.4</td>
<td>36.3</td>
<td>11.8</td>
</tr>
<tr>
<td>Subtotal Mini</td>
<td>87.4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Work Station Custom</td>
<td>79.0</td>
<td>33.9</td>
<td>13.3</td>
</tr>
<tr>
<td>Work Station Packaged</td>
<td>3.4</td>
<td>18.4</td>
<td>34.7</td>
</tr>
<tr>
<td>Subtotal Work Station</td>
<td>82.4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PC Custom</td>
<td>46</td>
<td>18.9</td>
<td>10.3</td>
</tr>
<tr>
<td>PC Packaged</td>
<td>14.5</td>
<td>3.7</td>
<td>12.0</td>
</tr>
<tr>
<td>Subtotal PC</td>
<td>60.5</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Thus localization of foreign packaged software and customization of almost all software are key industry and market parameters that have been created by the industry's historical
evolution and are now being perpetuated because of the high cost and organizational difficulty to users of change, combined with suppliers' strategic interest in maintaining their client base. This situation is persisting despite the fact that continuously customizing software is uniformly recognized by large users as quite expensive compared to the purchase of packaged products. Also, mainframes are seen as less flexible and responsive to change than network servers, workstations and PCs.

In this environment, the strategic issues have shifted from a producer's to a user's driven model. Yet, MITI's historical development role has generally been to support producers not to change user economics. Indeed, historically MITI's early software policies failed because the small software producers they supported developed products for which there was no demand. This experience does not bode well for the likely success of their new initiatives.

Because the clear user preference for customization extends beyond mainframes to mini, workstation and PC software, it can be argued that there is more involved than just the cost, risk, and organizational disruption of converting mainframe software systems to client server networks. While MITI and some developers see software as an industry to develop as an extension of MITI's inter-industry development model (Rapp 1993) and to support Japan's strategic computer initiative (Feigenbaum 1983; Anchordoguy 1989), users are only concerned with their industries. They are interested in the contribution information systems can make to their competitiveness in the market for steel, automobiles, consumer electronics, banking, insurance, etc. For them software is an input not an output.

Further, the technical contribution of the input is not uniform. Some programs are information processing, such as word-processing and spread sheets, while others are information
producing, such as CAD/CAM or super computer software. In the former case, the quality of the
information produced is largely independent of the program. A paper is good or bad whether it is
produced by pencil and paper or a sophisticated word processing program. In this case, Japanese
users see little benefit to supporting a Japanese produced program if it is more expensive or less
useful. In the case of information producing software, users actually see a potential disadvantage
unless the software is better than what they can obtain from the market. It is only the output,
either physical or as technical information, that users are interested in controlling.

Viewed from this perspective, important aspects of customization come into focus. As
described above, leading Japanese corporations have usually become globally competitive by
importing a technology and then improving the product and manufacturing process to achieve
higher quality at lower costs (Abegglen and Stalk 1985; Rapp 1992, 1993). They have become
global leaders by emphasizing process technology and innovation over product innovation (Imai
1986). A key aspect of these superior production and processing technologies and organizations
are the tacit skills of workers and management embodied in the firms’ established rules and
routines (Rapp 1992). Computerized information systems are an important codification of these
routines and represent part of the firms’ core competencies (Teece, Pisano, and Shuen 1990).

Therefore, software customization has not only allowed integrated systems providers,
especially Fujitsu, Hitachi and NEC, to tie their customers to their products; it has also served
user interests. It has done this by formalizing the uniqueness of a firm’s competitive advantage
and making it impossible for competitors to gain access to the total information system. Having
selected this decision rule and seen it work, the firm logically continues to pursue it, as no one
wants to chance that a competitor might otherwise gain an advantage (Rapp 1992).
In addition, while the cost of this decision from a software viewpoint may be relatively expensive, for many users software cost is not large relative to total output. For example, in the case of Toyota, this cost per car is only around yen 10,000, or 1-1.5% of sales, very small relative to the importance of maintaining Toyota’s production and organizational technology advantage. The cost savings of using packaged software are not large relative to the productivity gains of customization. There is thus a symbiotic relationship between the “captive” user and the integrated systems supplier that encourages continued customization. In essence software is part of a Japanese firm’s competitive evolution from imported technology and product improvement to global competitor, including the gradual integration of suppliers and customers into an integrated network of close business relationships (Smitka 1990, 1991).

Yet, users still must keep their systems technologically current. Often this means using more advanced or specialized foreign software. However, users will accept a certain time lag in adoption to adapt and customize software to their systems and to make sure all the “bugs” are worked out. This again shows Japanese users’ process orientation and administrative heritage (AEA 1992). In adopting foreign software, there are three steps necessary to sell large users: localization to Japanese language and format, conversion to their mainframe platform, and customization to the user’s unique process and requirements.

While initially localization presented some problems for foreign software developers, the process has now been refined through experience for most Western operating systems and application packages. Indeed, such conversion can now be done relatively quickly and easily. Several companies like Apple, Microsoft, and Hewlett Packard have modified the operating systems they supply to software developers globally to facilitate the conversion of application
programs to non-Western languages. The time lag between launch dates of English versions of a new operating system or application program and its foreign language counterpart has therefore become narrower. In addition, since many new application programs are upgrades, the amount of new programming that has to be localized has been reduced, further facilitating the process.

To better serve customers, the major Japanese integrated systems producers have opened offices in the U.S. to both attract and facilitate conversion of new programs to their operating systems. In this manner, integrated system producers subsidize and assist the entry of the U.S. software industry into the Japanese market, simultaneously lowering costs and expanding the user base for U.S. firms. Japanese users and integrated producers maintain their strategic integrity by improving foreign developers' competitiveness. In the process, they undermine MITI's attempt to develop an independent globally efficient Japanese software industry.

Several foreign software developers have taken advantage of this dynamic to further improve their cost competitiveness by developing a hub and spoke strategy that maximizes market penetration. They first develop a software package that can be sold globally and is localized for Japan. The localized package is then adapted to each major mainframe and mini producers' operating system, often with the producer's assistance. Competition takes place within new but smaller growth segments at the hub, i.e. network servers, workstations and PCs as seen in the table above. But there is little competition to shift core clients from their mainframe systems or operating platforms so those market shares remain stable.

Visually, high growth segments are at the center of a hub and spoke diagram representing the industry structure while the stable mini and mainframe market is represented along the outside of the wheel. The integrated systems producer's strategy is to maintain the customer base
by supplying hardware and software for all segments (mainframe, servers etc.) to existing customers. In this way, each such producer and system represent a spoke, with the mainframe system found at the intersection of the spoke with the outside rim. The independent software developer then tries to sell into as many of these client bases (spokes) as possible through adaptation, generally with the cooperation of the integrated systems developer. At the last stage, the software is customized for the individual users. Most large users do the customization themselves through captive software development subsidiaries to maintain confidentiality.

As might be expected from the above analysis, the new high growth software market segments in Japan are dominated by foreign software with high performance characteristics, sometimes incorporating downsizing and open systems. In work stations, HP and Sun Systems have more than 50% of the market. Yet, customization persists, requiring localization and conversion along the spokes, and Sun and HP have structured themselves to do this.

Users’ desire for open systems to facilitate communication among multiple platforms within firms or company groups also fosters market penetration by foreign software. Users adopt a multi-vendor purchasing system to control costs and avoid excessive dependence on their mainframe suppliers. Also, for certain uses, particular hardware-software combinations are superior, e.g. UNIX and engineering workstations or data bases and IBM compatible mainframes. Interviews suggest that firms such systems integrated with their customized information networks.

Yet, communication between firms or with networks using different platforms is necessary, it is not as frequent as communication within the firm and within particular functions. Thus, external communication need not be as high speed or as efficient as that taking place within a function or firm. This means that integrated systems producers can satisfy client requirements
and still maintain a captive system strategy by optimizing communication between its machines and software vertically but providing UNIX communication capability between network servers. This technical strategic mix is frequently achieved by having each machine type use the same family of RISC (Reduced Instruction Set Chip) microprocessors which are optimized for the integrated system producer’s unique operating system and UNIX.

FUTURE EVOLUTION

Further contributing to the software and computer hardware industry’s evolving hub and spoke structure is users’ increasing preference for the three tier computer system referred to above. Because of their huge investment in customized mainframe software operating on a highly differentiated operating system, large users are not converting substantial parts of their current software systems to network servers and open systems. Rather they are doing this at the margin for new programs and applications. Yet, they want to move in this direction and to operate a multi-vendor system. Thus, they have generally established systems where mainframes continue to manage large data bases but manage a series of network servers or office computers as well. These office computers provide the UNIX link to other machines as well as managing a series of PCs and workstations. Under this system, if work can be handled at the workstation or PC level, it stays there. If it needs to be communicated or shared it goes through the network servers, and if it needs to access or add to the data base it goes to the mainframe.

Since it will take several years for most large users to achieve full implementation of this system—and even at the end, 30% to 50% of the work will still be done by mainframes—customization will remain the industry’s basic characteristic for some time. This means a hub and
spoke marketing structure will persist as well.

Japanese integrated systems producers will continue to push towards joint ventures and exclusive licensing arrangements with foreign firms in new software technologies and formats to try to control access to their core customers to maintain market advantage in total systems and hardware. Foreign suppliers can use these motivations to increase their Japanese and global user base. In this way, the hub strategy pursued by the foreign software suppliers and fostered by the established computer hardware and operating systems suppliers increases foreign firms’ already considerable competitive advantages in the Japanese market. This conclusion is confirmed by the survey results. Except for games and Japanese language word processing, most users could not think of a single leading Japanese supplier of packaged software.

This leaves the industry, including the large number of captive software suppliers, to focus on maintaining technical parity for large end users through localization, conversion, and customization of foreign software and on developing additional applications for the existing systems. This industry structure thus relies on foreign development and a relatively stable client base and, except for games, there is little independent software R&D. At the same time, decompilation becomes important when a foreign supplier will not adapt its product to a particular integrated producer’s platform. Overall, the entry and presence of foreign vendors in the Japanese market will continue to be subsidized, hurting the development of local firms.

In sum, the presently fragmented operating systems and software application pattern will persist and standardization will not occur except for some work stations. Except for games and software incorporated in competitive manufacturing systems, internationally competitive packaged software is unlikely to be developed in Japan unless users’ desire for customization
changes. But there is no economic or management incentive for this to happen given user firms’ high sunk costs and strong desire to support their unique competitive operating advantages. In the future, the Japanese software industry will likely be composed primarily of profitable niche players, for example those producing Japanese language word processing and accounting programs or those affiliated with foreign firms.

Yen appreciation only intensifies the hurdles independent developers face. In fact, major manufacturers’ ability to maintain long-term competitiveness despite an appreciating currency means Schumpeter’s creative destruction is not occurring in Japan on an inter-industry basis. Therefore, economic and business pressures are not driving “advantaged resources” (Porter 1990) to seek new opportunities in areas like software as occurred in the U.S. with the decline in steel, autos, and consumer electronics.

The above analysis suggests that predictions that the U.S. software industry faced “inevitable decline” due to Japanese competition will not become reality. In fact, Japan’s phenomenal and continued global competitive success in other industries has undermined the competitiveness of its independent software companies.

In addition to strong user interest in customization, management practices also adversely affect Japanese software producers’ competitive position. Since the first goal of a Japanese corporation is survival, major systems producers like Hitachi, NEC, and Fujitsu will protect their core business even to the extent of offering loss or break-even services (Rapp 1994). They may even give away consulting or software services, as in the bidding scandals involving local government computer contracts or, less spectacularly, as in their willingness to facilitate localization of foreign software to their systems (AEA 1992). Having secured their base
businesses, system producers look for growth opportunities in the same market, like work stations or PCs, in emerging markets abroad, like China, or in related markets, like multimedia and games.

These expansion strategies frequently involve affiliations with foreign firms to access foreign technology and expertise. Fujitsu's recent tie-up with Apple in CD-ROMs, Toshiba's joint venture with Motorola in MPUs, Matsushita's investment in Go Video, SONY's acquisition of Columbia Pictures, and Toshiba's investment in Time-Warner are all relevant examples.

"Competitive compulsion" forces competitors to follow suit (Yoshino 1968; Ohmae 1991).

Another practice that hinders software industry development is the permanent employment system (Abegglen and Rapp 1972; Abegglen and Stalk 1985; Pucik 1984). While permanent employment may have facilitated postwar growth in many manufacturing industries, with software the system favors customization over packaged software due to Japanese firms' ability to train and retain EDP personnel. The mobility of the American labor force has required greater standardization of both operating systems and application programs. Given that the competitive advantages of Japanese firms and management identified by Aoki (1990) and Cusumano (1991) do not exist in software, it is not surprising that MITI efforts to extend the successful hardware paradigm to software have not worked. The failure of industry support effort is today forcing MITI to look for a new approach. But it is not clear officials are taking sufficient account of user economics and business objectives, or the different structure of this industry compared to industries in which government promotion policies succeeded.

GAMES

Confirming the validity of the evolutionary approach over cultural or best practice theories
is the fact that in games, Japanese firms do not suffer from these adverse economic conditions and are global leaders. That is, the failure to develop a strong business packaged software industry is due primarily to user economics and strategies in combination with the industry’s evolution, not to management failure to adopt best practice or to cultural differences between the U.S. and Japan. This situation also confirms Baba et al’s conclusion (1993) that Japan’s software industry is heavily user-driven. Sega and Nintendo have leveraged a large, rapidly growing domestic market for games to adapt to the global market. This has given them a significant installed user base over which to spread their development costs, dramatically increasing their efficiency in revenues per line of code written. Further, the large household investment in game cartridges has acted as an entry barrier to other firms and a deterrent to users’ switching systems. Thus first-mover advantages during periods of technical upgrade are important, after which the firm can play a classic strategy of almost giving away machines (razors) and selling games at large mark ups (razor blades). This is analogous to U.S. software producers’ pricing new programs aggressively to get the subsequent upgrade revenues. U.S. packaged software and Japanese game industries thus share many common economic characteristics.

Continued fragmentation of the business software market and lack of a common user base emerge as the prime reasons for Japan’s failure to develop a large independent software industry. Nevertheless, in terms of U.S. public and industry policy, there remains a need to monitor MITT’s new development strategies and to keep the market for U.S. software open (Prestowitz, Morse and Tonelson 1991). U.S. managers should continue to exploit the hub and spoke strategy while learning more about the market from Japanese systems producers and large software users. U.S. policy should encourage cross-border alliances that assist in penetrating fast growing market
segments at the hub and should carefully assess the impact of specific actions like taxation, anti-
trust and intellectual property on such global competition. Managing the competitive process
entails pursuing corporate strategies and public policies that reflect past successes and defeats.

THE ANALYTIC RESULTS

Many leading Japanese manufacturers have out-competed U.S. firms worldwide in a
number of industries in which the U.S. was once a leader, with adverse consequences for growth,
employment and profits at the American firm. To explain this competitive development, a number
of studies have compared Japanese and American firms. Their detailed descriptions of variations
in management practices have been informative and supported various arguments on how these
differences result in competitive advantage for Japanese firms. They have covered a wide range of
industries and research approaches and include Abegglen and Rapp (1972), Anchordoguy (1988),

Studies like those by Womack, Jones, and Roos (1991), Krafkic (1988), or Dertouzos,
Lester, and Solow (1989) argue that Japan's advantage is in its process and manufacturing
technology. They believe lean production systems can be applied with competitive advantage to
any environment or industry. Others like Dertouzos, Lester and Solow note that Japanese firms
are not leaders in industries already using continuous processes such as chemicals, paper, or
refining. Semiconductor production involves a complex series of discrete and increasingly
sophisticated manufacturing processes subject to learning and economies of scale, which ideally
suit Japan's area of competitive advantage.

One implication of the excellent study by Cusumano (1991) on Japanese software factories is that software development can be organized similarly with similar results. The system described substantially reduces the time and cost of designing and producing an amount of software code while improving quality and reliability. This was expected to translate into a competitive improvement in Japanese relative to foreign software. One might call this expectation the Feigenbaum/Cusumano paradigm since Feigenbaum (1983) and Cusumano (1991) were the first to challenge the beliefs that the U.S. held an insurmountable lead in software and that Japan's competitive growth paradigm and lean production advantage did not apply to software. They noted that many elements contributing to the successful challenge in high technology electronics were in place for software. First, the firms involved have the same benefits of group resources, capital availability, permanent employment, leverage, joint industry government commitment, technical knowledge, etc. as in semiconductors. Secondly, disciplined factory-style management can significantly improve productivity and quality, with the amount and reliability of code produced exceeding levels achieved by U.S. firms.

Additionally, as in ICs, the Japanese government has targeted software for competitive development as part of its support for computers and Japan's thrust into the information age. This is seen in the Fifth Generation Computer (ICOT - Institute for New Generation of Computer Technologies), Sigma, and TRON projects. There is also the national High Speed Computer Program and Japan Software Development Program. Support has run in excess of $600 million. In all these programs, government and industry have cooperated, with MITI putting in funds and resources along with industry. Further, they have again tried to select the best people while
keeping the number of firms limited to capture whatever scale and experience benefits exist.

To overcome foreign firms' competitive advantage in trained and experienced software developers, the government and industry also tried to shift the production function from one dependent on highly skilled experienced people to a more capital intensive one where integration of worker and machine was more important. In this way, Japan's software developers could benefit from Japan's high saving's rate and demonstrated strength in lean production. Therefore, one stated objective of ICOT was to increase capability for machine-written programs and to integrate computer assisted programming into software production.

If this objective had been achieved, it would have helped create a beneficial competitive cycle for software where Japanese firms could develop a more continuous software production process with fewer defects. This would lead to better quality, lower cost software which would lead to more sales of software and hardware. This would then result in the kinds of synergism and quality improvements witnessed not only in ICs but also in many other Japanese hardware industries. In fact, success here would have created demand for such machines and their related programs, further improving production benefits. It would repeat the IC/semiconductor equipment/electronics scenario. However, this projected scenario failed because it did not recognize the difference in industry economics for software producers and users.

INDUSTRY ORGANIZATION, INVESTMENT, AND COMPETITION

Nevertheless, several firms have organized themselves along factory lines and have established disciplined rules and incentives for the repetition, reuse, and cross utilization of certain programs to reduce duplication. While initially pursued as a way to become more competitive, it
has persisted because it works as a way to reduce costs in developing customized software or adapting and localizing foreign software. That is, these procedures have facilitated continuous improvement in existing programs in code written per hour and their reliability and quality since improving the reliability of foreign and customized programs remain important objectives given the high cost of a system breakdown (AEA 1992). It also reduces the number of experienced software engineers required to produce a usable program, an important benefit since such technical personnel are in scarce supply. Indeed, the aging of the Japanese population means that there will be fewer technically trained people entering the labor force in the future which will increase the scarcity of skilled programmers and software engineers. By breaking the task down into discrete modules, people with less training or even contract personnel can do the work. Further, industry commonality and repetition of successful software designs and line codes are used as a way to reduce errors and improve output per programmer. This is analogous to building or designing in quality through constant repetition and continuous improvement as seen in semiconductors and other industries. This has been demonstrated, however, only for larger firms (Cusumano 1991). It is not clear if such results extend to smaller firms. Further, it has not extended to the competitive development of new software programs.

As seen in games, Japan has the domestic market size for an effective launching platform for its software producers. It is the second largest software market in the world and generally has grown more quickly than the U.S. market. Large firms or groups can still subsidize growth via leverage or the stock market. Thus capital availability remains an advantage even though low interest rates in the U.S. and a good IPO market have done much to eliminate this advantage.

In the early 1990s, there thus appeared several similarities between software production
techniques applied by large integrated electronics producers like Hitachi and Fujitsu and the ones they used to produce ICs. Managers logically acted on the basis of past success. Government also supported the industry using similar cooperative development programs but with a policy emphasis on common standards, concentrating resources, and high market shares, all proven techniques to reduce cost and improve quality among select firms which then spread to the industry, furthering overall competitiveness. However, due to the many important differences between ICs and software in terms of Japan's success model identified above, especially user base economics, these same managers have now shifted to a hub and spoke technique and are adapting foreign software to their clients' proprietary software systems. Because of this shift in perspective and because it would also necessitate substantial program revisions, TRON has not been accepted by Japanese industry or by countries outside Japan as an alternative dominant design to operating systems already on the market. This has frustrated Japan's development of an independent software/MPU system and a full shift to UNIX. But it is also rational, because without acceptance abroad, TRON cannot gain the benefits of a global user base, and without those benefits, it makes little sense for domestic firms to adopt it since it would be high cost and of limited use. Rather, it would mean developing new localization and adaptation programs to customize it to their operating systems, or if everyone were to use the same system and programs, it might mean loss of their proprietary production systems. Both these choices are unattractive to Japanese users compared to the existing situation. Therefore, they have not supported it.

In looking at successful Japanese competition in industries like semiconductors, autos, and consumer electronics, one theme that is constantly repeated is managing growth as the key to increased market share and firm survival rather than relying on labor mobility and profits. This
emphasis seems most successful in industries where a standard product is produced on a repetitive basis involving hundreds or even thousands of parts and using complex production tasks where the opportunities for continuous integration are great. However, some benefits from these factors relative to software development and competitiveness appear questionable due to the differences observed above between software and semiconductors in industry structure and competitive dynamics. The role of timing in this, though, is also important. U.S. firms retained their lead in semiconductors as long as U.S. growth was high. Once initial growth slowed and firms became more subject to cyclical pressures, Japanese firms could use their greater resources, including keiretsu and government support, to chase a more consistent target even in economic downturns. Could a similar situation emerge for software when growth slows?

The market and technology are closely intertwined. As long as a product is successful, one can finance the next, but if growth slows, it becomes more difficult. So firms cut back, giving Japanese companies an opening. Further, it is usually when the market slows and matures that technology begins to stabilize, standards emerge, and it is easier to mobilize resources to catch up. However, it also becomes more difficult to shift market share and costly to unseat established leaders. Therefore, because the mainframe market in Japan is mature, market fragmentation is likely to persist for mainframes, which given the need for system integration forces customization in the new high growth segments like network servers and work stations as well. Further, there is a higher degree of foreign software participation in the new high growth segments due to the success of their hub and spoke strategy. So it does not appear that Japanese producers could easily leverage their domestic position into an aggressive expansion strategy during a downturn, while their localization skills are not necessarily reversible to develop products salable in global
markets where U.S. firms appear to be capturing the growth. Indeed, systems development for international sales has proved difficult for Japanese firms—even in areas like cellular telephone systems where they have competitive hardware—since Japanese operating systems have not easily adapted to overseas billing and record keeping requirements.

Emergence of a standard or dominant design might favor Japanese producers, but it must be demanded by Japanese users. Unfortunately, there seems to be no strong trend in this direction for software. Rather, economics and administrative heritage suggest customization will persist. In consumer electronics, the effects of multimedia and giant global networks may force standards for these products, and Japanese firms should be well positioned to benefit from this development no matter how it evolves. Yet, they will find plenty of competition from the Koreans and a resurgent American industry. Further, the operating and application programs are likely to remain American, though Japanese firms will continue to do well in games.

Such administrative heritage is a major reason for the persistence of customization. This can be clearly seen in the steel industry, where some of the earliest software developments for operating systems were pioneered. Even today several of Japan’s leading software houses are steel company affiliates that spend their time developing software for the steel industry (Baba et al 1993). It is in the production systems of such leading and successful global competitors that one sees the results of Japanese software innovations and developments. Toyota’s vaunted production system has highly integrated computer systems that begin with the order from the customer and then schedule the production run, notify the appropriate suppliers, etc. In this respect Japan has a very successful and competitive international software industry, but its competitiveness emerges in a car rather than a shrink-wrapped software package.
Seen in this light, leading Japanese firms' emphasis on customization makes perfect sense since software is part of a successful integrated production system. Software efficiency and information processing technology are realized not in software production itself but in the production of steel, ships, etc. Solution programming expertise comes not from localization and hardware adaptation but from the software's customization and integration into the customer's existing system. In turn, the cost of the software in a ton of steel or a car is not great enough that it would pay to buy off-the-shelf software if that meant sacrificing the benefits of a firm's unique production process (Imai 1986). Buying existing software that has to be localized, customized and debugged may not be that expensive when compared to developing a system from scratch.

This is a major reason why Japanese firms have maintained a strong mainframe mentality, allowing the market for mainframes and their software to persist more strongly than in other developed markets. Large users want to stay close to existing technologies, products, and ways of doing business since this entails less risk. There is rarely a benefit in being aggressively innovative in Japan: experience indicates that slow incremental improvements over a long period of time is the way to achieve competitive advantage, beginning with what one knows will work (Imai 1986). In this environment there is little user support for leaping technical frontiers in software. When the heads of users' EDP departments are not EDP specialists but come from the business side, this further retards dramatic change. EDP managers have little interest in pushing EDP innovations; instead they are concerned about the impact of such change on the organization. The benefits of major changes in software systems are uncertain, especially when Toyota and others have demonstrated that the most computer-intensive production systems are not necessarily the most productive or cost efficient. The most automated production system in Japan, Nissan's Zama
plant, has been shut down in the current recession, and GM's highly automated facility at Lordsville was not a success either.

In this case the arguments of economists like Aoki (1988, 1987) that Japanese competitiveness flows from its organizational differences seem applicable on an intra-industry basis but not on an inter-industry basis. The organization of Japanese industry has actually frustrated the development of a globally competitive software industry while it has promoted successful customization of software to the needs of its leading software users. Still, even among major users, there is much room for improvement beyond the shop floor. Compensation and promotion based on seniority have retarded adoption of computers in the office, reducing the productivity of white-collar workers. Management feels no competitive pressure to introduce newer software technologies.

This differs greatly from the situation in the U.S. where job mobility is high and where computer literacy can now be an important executive requirement. There is some evidence that mobility among Chief Information Officers and other EDP experts over time has been partly responsible for the greater standardization of operating systems and application software in the U.S. This in turn has supported the evolution and growth of packaged software where job stability promotes customization in Japan. Interviewees in both Japan and the U.S. agreed that the adoption of newer open and more flexible software systems without major changes in organizational structure or motivation was unlikely to be successful. Such large scale organizational change seems to occur more readily in the U.S.

Maintaining U.S. competitiveness in software appears to depend on the continuation and promotion of certain management practices and institutional arrangements. While these practices
and arrangements worked to the advantage of Japanese firms in many industries, in software they seem to benefit U.S. firms. This counters Aoki’s conclusion that management success is not industry specific. Ironically, it is the importance of cooperation within firms and between Japanese and foreign firms in achieving their overall business objectives which is partially responsible for this. That is, if tacit knowledge transfer (Florida and Kenney 1991) is critical to the competitive success of Japanese manufacturing routines and customized software is an important codification of those routines, the software industry’s fragmentation will persist. In this way, observing differences in current management procedures as extensions of the origins of those practices and the motivations behind them has helped the researcher team understand the competitive dynamics and probable future development of the industry.

Japanese software producers have found it much more difficult to copy foreign software technologies than have producers in industries like semiconductors (Rapp 1993A). This is a key aspect of the frustration of MITI’s software industry promotion policies. Addressing this problem may require Japanese software systems engineers to train in the U.S. to achieve a transfer of skills, just as Japanese manufacturers in the U.S. have had to work diligently to transfer skills to their American staffs (Florida and Kenney 1991). However, given that U.S. firms are concerned with technology loss and do not necessarily need to build new linkages with Japanese systems engineers to open the Japanese market, it is unlikely this will occur on a large scale.

SOME CONTRASTS WITH U.S. INDUSTRY

Further differences with the U.S. are that U.S. EDP departments are generally run by people with strong EDP backgrounds who see their careers as EDP specialists rather than as
generalist managers. This situation has been reinforced by U.S. labor mobility combined with the industry's pioneer status. Users initially hired technical experts from systems developers and then from other large users. Further, the development of systems integrators and software producers like EDS and Microsoft meant one could rise to the top of the corporate ladder and achieve great wealth solely within the software industry. Thus in the US, a career as a software specialist has not only been possible but attractive (Steinmueller 1993). No similar opportunities have existed in Japan, except perhaps in games. Rather, software was fairly well developed when it entered Japan in the 1960s, and entry was within the normal corporate context.

Since U.S. EDP specialists in corporations often viewed themselves as computer specialists first and corporate executives second, they looked to the computer industry to facilitate their mobility. When they shifted from one firm to another, they did not want to have to learn a new set of technical skills in addition to learning about a new company. Further, the corporations hiring them wanted them to be productive quickly, favoring standardization. Universities that used computers and offered courses in computer science were caught in the same push for standardization and career mobility. These institutional, economic and management forces promoted further integration, while similar forces in Japan encouraged fragmentation.

Given the technical expertise necessary to run large mainframe computers during the industry's early development, careers in the computer industry were facilitated by users and the lock-in effects to a particular standard technology and supplier as technicians moved from user to user. Suppliers promoted this mobility, training and career development as it helped secure their own markets long term. At the same time, this pattern suggests why many corporate EDP departments were reluctant to embrace PCs until the introduction of networking. Many MIS
specialists saw the migration of computing power and expertise to operating personnel as a threat to their technical monopoly within the large corporation and thus their power and budget base. This anxiety continued until it was clear that they were still needed to provide the extensive information sharing networks within and between organizations needed to access large mainframe data bases. In addition, the increase in computing power made possible very large systems requiring their sophisticated expertise. Even in the U.S. there is a hard core of uses requiring mainframe capabilities into the foreseeable future.

In fact, in the U.S. the availability of generic programming and of relatively inexpensive prepackaged solutions has meant organizations have had to do more to make this software work for their particular situation. Under these circumstances it is hardly surprising that the number of programmers and EDP expenditures within users in the U.S. is almost double that for systems integrators, service organizations, and prepackaged software (Steinmueller 1993). Conversely, in Japan, where software providers are often affiliates of the user and customization is the norm, direct internal user expenditures and personnel are smaller and are roughly equivalent to that for software systems houses. This situation is strengthened by the fact that software providers in Japan often rent programmers to firms under contract for specific projects (Baba et al 1993).

Thus in Japan and the U.S. software employment appears to be a mirror image. Similarly, the reengineering phenomenon sweeping corporate America seems to be an attempt to achieve a better integration of a firm's basic business, operating personnel, and information systems, as already is the case in Japan, while Japanese government policy wants to move more towards packaged solutions. This U.S. trend is driven by the fact that many firms feel they have not achieved the expected productivity increases for their systems investments (i.e. computers and
software). Frequently this has been because the systems do not reflect the basic customer or firm interface and are not user friendly to employees. Partly this is due to these firms’ EDP heritage noted above where purchase decisions were made for the convenience of the EDP specialists running these activities, not because they were suitable for the business. In addition, once in place, those systems often drove the software solutions selected for other business or information needs. That is, systems selections were made for reasons of compatibility with existing machines and systems, not because of their ability to efficiently provide what was required.

As information processing has become necessary for the basic business, however, and as computing power has been distributed more widely to operating personnel to use in their customer interface, the inadequacies of a non-integrated system have become more apparent and increasingly less tolerable. In fact, when a firm’s competitors have been able to significantly improve this integration through reengineering, the productivity results have frequently been amazing and have forced other firms to respond. IBM’s solution marketing is clearly a response to this, while MITI is hoping such competitive compulsion will force Japanese firms to move towards less customized solutions and more open systems.

In Japan, though, customization and integration have been the norm. While this has made Japan dependent on foreign technical advances in software and computer processing (logic), it has not affected firms’ ability to utilize such inputs for their competitive advantage. This is why users have little interest in supporting MITI’s attempts to move software development towards the greater commonality. Interviewees stated very clearly that the government’s push towards open systems had no affect on their software decisions, which uniformly supported continuation of customization. Nor does any reversal of this trend seem likely. The current joint ventures and
cooperation between U.S. software developers and Japanese customizers cater to large users’ continued desire to customize while helping foreign firms build their user base. In addition, globalization of Japan’s leading industries have not increased the need for integrated operating systems. Except for certain production systems, headquarters permits foreign operations to determine local software needs based on the local environment, with only data interchanged between the parent and the foreign subsidiaries.

THE POLITICAL ECONOMIC DIMENSIONS

Historically, an important research issue has been that successful Japanese international competition has often had a political and public policy dimension and these considerations have clearly extended to software. MITI’s involvement in software development initiatives past and present, and their support of Japan’s computer industry have logically extended the investigation of the industry into the political economic sphere. That is, the research team recognized that Japanese MNCs have been influenced in their evolution by Japanese government policies and outside political economic pressures. This has been true in computers as well as in major user industries such as steel, autos, semiconductors, satellites and machine tools. Further, Japanese trade, investment and industrial policy have been the topics of extended and often acrimonious negotiations with the U.S. government.

While bilateral negotiations have had a significant impact on Japanese firm behavior in a number of key industries, in computer software such talks have been less an issue, with the exception of negotiations over intellectual property rights. In part, this has been because major Japanese software users have increased their independence from the government (Rapp 1993).
When questioned about the government's influence on their software decisions, all large users responded "none". Similarly, large users said that MITI had no ability to force them to buy domestic software as the ministry had done with domestic computers (Anchordoguy 1989). In sum, as Japanese firms have become more competitive, financially stronger, and globally expansive, the authority and "coordinating" role of MITI and other government ministries has been viewed as weaker. This has benefited foreign software developers.

SUMMARY AND CONCLUSIONS

This paper has shown the superiority of the evolutionary model over the adoption of best practice model or the culturally determined model as the best explanation of the current state and projected development of Japan's software industry. As the evolutionary paradigm would predict, the research demonstrates that even leading Japanese firms have some difficulty in adapting to the new global competitive conditions. Rules and routines that produce success in some industries cannot necessarily be applied to the software industry with the same success. Japan is quite competitive in certain software segments: graphics, games, and certain multimedia applications. However, in most other segments the bulk of Japan's technical and financial resources are not devoted to developing software as an output, but rather to customizing systems that support leading-edge manufacturing firms as an input into the production process. Localization and adaptation dominate the corporate software market in Japan.

Evolutionary theory maintains that certain corporate rules and routines develop in response to situations or events that often have little to do with their later benefit to the industry or firm. These practices then evolve over time in response to changes in economic, industry, and
firm circumstances, bringing more change. In essence, corporate history and administrative heritage matter. Under evolutionary theory, the firm or industry is viewed as unique, albeit with some common rules and routines across successful firms. A key objective of this paper has been to identify these common and unique elements in Japan's software industry as well as in specific market segments. Most importantly, the research has identified the emphasis on customized software, driven by what firms see as their competitive advantage, as a critical characteristic of the industry, contributing to a high degree of fragmentation.

Yet Japanese firms and their competitive challenges are moving targets. Despite favorable evolutionary developments to date, foreign software developers need to keep in mind the agendas of Japanese customers and major systems integrators as they assess the unfolding competitive scenario in software. Japanese goods producers have invested aggressively in software and other productivity enhancing technologies to lower costs in the face of the yen appreciation and heightened competition from newly industrialized Asian economies. So far, however, this effort has primarily involved adapting and customizing foreign software. Any change in this pattern would be a cause for concern for U.S. software developers: if the preference for customization were to change, so would the market for software.

The key industries in this regard are those in which Japan has shown particular competitive success—steel, autos, electronics, machine tools and their related industries. They are the source of considerable software demand and can pull Japanese software into the global marketplace. Japanese software developers for these industries could create a customer-driven global market advantage contributing to the production integration typical of these firms, if the degree of customization could be reduced.
Such conclusions must accommodate the possibility of a change in government policy and corporate strategy as MITI and other leading players respond to the lack of success in past initiatives. Computers and software are a global business in which firms naturally want to maintain or improve their competitive position. Recent initiatives by major consumer electronics and semiconductor producers in the video game software and MPU markets clearly indicate that the current competitive lineup should not be assumed to be static. Reliance on cultural impediments to Japan’s emergence as a major global software competitor (Delaney 1994) is not realistic, as the situation in games demonstrates.

Whereas the consumer electronics industry has supported demand for Japanese IC producers, the comparable products for MPUs have been work stations and personal computers where US producers have established a strong global position, including in Japan. U.S. PC and workstation manufacturers are successfully penetrating the Japanese market by aggressive pricing to build market share (AEA 1992). They are helped by a strong yen and new U.S. originated operating systems that are more user friendly in Japanese. This development adds to the self reinforcing negative cycle for Japanese PC, word processing and workstation manufacturers and their affiliated software producers. While multimedia may change this competitive dynamic given Japanese firms’ strong position in consumer electronics, the present situation is so fluid as to offer no clear direction. Further, the spillover from multimedia to PCs and work stations is unclear.

Also, the fact that software must be learned on a particular hardware platform increases the importance of an installed software and hardware base in attracting programmers to develop new software for that base. This has given significant first-mover advantages to U.S. software producers. For example, American personal computers generally are either DOS or MAC-based
(and Windows builds on DOS to make it look more like MAC, indicating convergence at the user level). In turn, the most popular MPUs for PCs support these software platforms, e.g. Intel's 286, 386, 486, Pentium and compatible chips or Motorola's 8030, 8040 and Power PC. Users' compatibility demands have forced commonality, further helping Intel and Motorola as computer manufacturers demanded their chips or chips compatible with their designs.

This creates a self-reinforcing cycle as programmers lower their risk by developing programs for these platforms. This situation sharply differs from textiles, steel, automobiles, TVs, and DRAMs where one can freely enter an existing infrastructure and supply a product which can be used interchangeably with products of other vendors. The need to ensure compatibility has led to alliances like that between Toshiba and Motorola, where each firm accesses the other's strength in process and design. The ultimate beneficiaries of these arrangements are not yet clear, but it appears that Japanese adaptation and manufacturing strengths have not predominated.

At the same time, U.S. firms like Intel and Motorola have set up MPU production in Japan and have gotten much tougher on licensing and reverse engineering, limiting access to their technology. Now, as production growth in Japan for ICs shifts toward the NICs, American firms retain significant market share in MPUs as well as a Japanese production and marketing base. This improves their competitive position in software and their hub and spoke strategy.

MITI is presently trying to shift the nature of government support for Japanese non-game software developers, hoping to promote small and medium size companies, whose software seems to be in demand. However, there seems little difference in strategy from the one that failed in the late 60s and early 70s. The ministry still seems to be trying to treat software as an output industry rather than as an input into other industries. Trapped by past successes in output industries, MITI
has not fully integrated the unique economic, competitive, and political features of the software industry into its plans. The government is no longer in a position to force users to accept or subsidize locally developed software if localized, adapted, and customized imported software is cheaper, works well, and is more technically advanced. A better approach would be to go with the economic and competitive flow, promoting industry-enhancing software development to assist major Japanese users in maintaining their global competitiveness. This would in part involve the export of sophisticated manufacturing and process software systems as user firms make investments overseas. Otherwise the government's options appear quite limited.

As long as software is technologically advancing and changing rapidly, U.S. producers seem at an advantage. They can more easily maintain this pace in software because development is less capital intensive than ICs and the primary input, skilled people, is fully tax deductible. Thus the capital cost issue is less of a benefit to Japanese companies. Further, U.S. venture capitalists have actively supported the market, partly because they perceive Japanese and other foreign competition as weak. Finally, because capital is cheap in Japan and readily available to large firms, major Japanese computer firms can, as noted above, support a range of software firms related to their hardware platforms. This has prevented a more rapid rationalization of the market.

As there is no dominant global software design across all segments, potential economies of scale from dominating a domestic market have not proved extendible to global markets, with the exception of games. This has frustrated further development of the beneficial cycle of sales, scale, and price witnessed in semiconductors. Indeed, such domestic dominance when built on a strategy of perpetuating a unique market position based on an incompatible standard has created problems for firms like NEC in personal computers. Efforts to maintain dominance of the Japanese market
has forced it to spread its software resources across two operating systems. However, producing fully compatible machines would eliminate much of their lock on the Japanese PC market: it would become too easy for customers to switch to other producers as they upgraded to new machines. Further, as their PCs can communicate with their larger computers, such switching could affect these sales as well. The two-operating-system strategy is a little like riding two racing horses simultaneously, hoping that you can successfully jump to the right one at the last minute.

In sum, the presently fragmented operating systems and software application pattern will persist and standardization will not occur except for some work stations as firms move towards UNIX and open standards. Further, except for games, locally developed, internationally competitive packaged software is unlikely to occur unless users' desire for continued customization changes. But there is no economic or management incentive for this to happen given users' sunk costs and strong desire to support their unique competitive operating advantages in their basic businesses, which are not software. Integrated system suppliers also have an incentive to maintain the present system, which keeps users tied to them.

The evolutionary theory predicts that even leading Japanese firms can have difficulty adapting to the new global competitive conditions in software, and this has occurred. Indeed, the continued success of certain corporate routines in their basic businesses has made adaptation to a new set of circumstances more difficult or has led them to stress particular areas of competence. Evidence presented in this paper allows prediction about which software segments are favored by Japan's competitive strengths. They are graphics, games, and certain multimedia applications as well as the unique customized systems supporting competitive prowess in industries like steel, autos and consumer electronics.
Japanese goods producers have invested aggressively in software and other productivity enhancing technologies to lower costs. So far, this has been primarily foreign software, adapted and customized to their needs. Any change in this situation should cause concern, as success in games indicates that reliance on a cultural impediments view to Japan’s emergence as a major global software competitor is not realistic. Failure to recognize the economic and organizational impediments due to industry’s evolution would thus be a policy error. This understanding will hopefully help the U.S. and Japan to better manage the realities of the global software market and the continuing political friction surrounding their competition in high technology.

REFERENCES


Teaching Note #5-388-103.


____ 1989. "Motorola and Japan (A)." Teaching Note #5-389-008.


