FINAL REPORT TO JAPAN-US FRIENDSHIP COMMISSION

THE FUTURE EVOLUTION OF JAPANESE-US COMPETITION IN SOFTWARE: POLICY CHALLENGES AND STRATEGIC PROSPECTS

GRANT NO. 94-46

Submitted by the Center on Japanese Economy and Business
Columbia Business School, Columbia University

August 31, 1995

Grant Period: June 1994 to May 1995
(Second year of a two year renewable award)

Project Directors: Professors William V. Rapp and Hugh T. Patrick

Center on Japanese Economy and Business
521 Uris Hall
Columbia Graduate School of Business
Columbia University
New York, New York 10027

TEL: 212-854-3976
FAX: 212-678-6958
SUMMARY

This document serves as the final report covering the Fiscal Year 1994-95 activities by the Center on Japanese Economy and Business ("the Center) at Columbia University under Grant No. 94-46, "The Future Evolution of Japanese-US Competition in Software: Policy Challenges and Strategic Prospects," pursuant to Article III of the Grant Agreement.

In June 1993, the Center on Japanese Economy and Business was awarded a grant from the Japan-United States Friendship Commission for the first year of a two year examination and analysis of the strategic, competitive, and policy relationships between the Japanese and US computer software industries. The grant was renewed in June 1994 for the 1994-95 grant period. The following activities were conducted with JUSFC funds during this latter period as well as with funds from the Center on Japanese Economy and Business, the Center on International Business Education and Research at the University of Washington, and the Centre for Asia Pacific Initiatives at the University of Victoria. In addition, we are submitting a final report which presents most of the major findings of the study. The findings are based on interviews and meetings with industry technical experts, other analysts, government policy makers, large systems houses, large software developers and large customers. These conversations were in turn supported through the use of questionnaires. The research program also included a review of existing data, reports and studies.

The study has focused on large systems producers, large software developers, and large organization customers because large customers currently account for most software costs and usage. Customized software for such clients represents about 85% of the total software market, and an even higher percentage of large customers’ expenditures. Software for mainframes and minicomputers used by larger customers represent over 80% of the Japanese software market.
Further, most PCs are used in offices not in homes. So even the sale of prepackaged PC software is heavily dependent the purchase and usage decisions of large customers. This situation is expected to change only very gradually. While this final report is complete in itself in its analysis of the material gathered for such large producers and customers, we expect these conclusions and the information assembled will provide the foundation for further work and analysis in the months ahead. This additional work is expected to include a review of our conclusions by other analysts, a book on the Japanese Software industry that will address many of the issues raised by our research and indicated in our final report, and an expansion of the simulation model.

The primary initial purpose of the research project was to understand the strategic forces impacting the Japanese software industry and whether as an industry targeted by the Japanese government it was likely to replicate the evolution of the Japanese semiconductor industry. The policy concern was that though foreign firms and especially US firms have about 75% of the Japanese packaged software market, this might prove temporary as Japanese firms with government support allocated more resources to this industry. There were several studies that hypothesized Japanese production methods were applicable to software and would prove quite competitive. After studying the situation under last year’s grant, the project team reached the initial conclusion that the Japanese industry’s development was heavily influenced by its historical evolution and that large customers together with large integrated systems producers were pursuing software strategies that would continue to emphasize customization and that in turn would promote industry fragmentation. This fragmentation together with economic and technical trends in Japan and the rest of the world would maintain foreign dominance of the Japanese packaged software market. Confirming and verifying these working hypotheses as well as further investigation of the industry’s current development was this year’s research focus.
During the grant year, the project team

1) accumulated and reviewed additional literature on the US and Japanese software industry. Further interviews and discussions were completed with industry, government, and academic experts familiar with both the Japanese and US industries. These interviews were then used to format and revise the Japanese and English questionnaires that were distributed to industry participants. The results of the questionnaires were in turn examined and related to the results of the interviews and the analysis of other studies and data on the US and Japanese software industries.

2) Based on these research inputs, the working hypotheses that were developed during the first year of work were confirmed and extended. These conclusions were disseminated and presented at various forums in order to solicit additional analytical comments and suggestions.

3) To confirm the working hypotheses, executives of large Japanese firms in several industries were interviewed with respect to their software usage and development plans, including their views on Japanese and US software technology. These interviews followed a specific questionnaire format. One of the working hypotheses that was addressed through this process was that American software customers rely relatively more on packaged software which is less expensive while Japanese firms continue to rely more on customized software, and even when buying packaged software, heavily customize it to conform with their unique and proprietary software systems.

4) Several Japanese software developers and systems suppliers were interviewed, and questionnaires were sent to others, following a similar and related questionnaire format to that provided to large users.
5) The interviews and additional research data have confirmed and extended the original working hypotheses concerning the reasons for the industry’s fragmentation and why this is likely to continue. The results have also shown that Japanese firms for various reasons are institutionally committed to certain rules and routines with respect to software usage and development that appear very difficult to change even when they are not optimum in terms of current software and computer technology. In most cases, this decision relative to software usage and development appears to be related to a firm's commitment to other routines as a way to maintain its competitive advantage in its own businesses. This examination of the origins of Japan's management practices as they relate to software seems important to understanding the future of the US and Japanese software industries since other competitive or use considerations than just the price and quality of the software itself often determine software purchase or usage considerations. These factors are frequently historically based. Thus the evolutionary approach has yielded important empirical and analytical results. It may also be possible through the simulation model to use this data to implicitly quantify the value to a user of maintaining such proprietary software technologies compared to adopting more general packaged software solutions even when the latter is cheaper and technologically more advanced.

6) An initial policy research concern in terms of the original proposal and research agenda was the degree to which the strategies, rules and routines that succeeded for Japanese corporations in certain industries were applicable to computer software. The global competitive success of many Japanese firms made this a major policy concern. However, the results of the project’s research findings indicate that though computer-related software is an essential input into virtually all forms of manufacturing and services, in terms of cost it is usually a relatively small percentage of the total cost of producing the product or service. It is therefore the economics of those
businesses rather than the economics of the software industry that determines large
organizations’ demand for and usage of software. In turn, those economics continue to stimulate
large customers’ extensive use of proprietary software systems to maintain competitive
advantage. Therefore, Japanese software suppliers have not been successful in transferring from
hardware to software Japan’s successful corporate production practices based on the
continuously improved production of standardized products.

7) In contrast to the 1960s and 70s, currently most large customers are buying software from
multiple vendors. Large integrated systems suppliers now find that only about 10% of their
customers buy only from them. Thus at one level the issue of group affiliation appears less
important in software usage than in some other industries. However, most large software buyers
have also created software development subsidiaries as a way to centralize their management and
cost controls over software use and development. These subsidiaries are thus part of these
parents’ vertical keiretsu. They in turn try to reduce their development costs per software system
developed for the keiretsu while maintaining their tacit expertise within the group by selling their
customized software to other members of the group. Such software development subsidiaries and
affiliates are among the very largest software companies in Japan and serve several functions. In
addition to helping control costs, they offer careers to specialized EDP personnel outside the
parent firm's personnel system. They also expand the firm’s software user base to reduce the
overall cost of maintaining a proprietary software system. The multi-subsidiary as opposed to the
multi-divisional approach to Japanese corporate organization of course has a long history in
Japan and appears to be a preferred organizational form when, as in this case, it meets firms’
basic strategic objectives.

8) While most large customers buy their operating and middleware systems from hardware
vendors or specialized software developers, most develop their own proprietary application systems either internally or through their software development subsidiaries rather than purchasing standardized packages off the shelf. Even when application packages are bought, they are often extensively customized. The questionnaires and interviews indicate that the cost of this semi-customization usually runs about twice to three times the cost of the basic package. This situation has forced most software developers and systems integrators to specialize by industry. In turn, their dependence on specific customers in particular industries, each with their own large proprietary systems, has made software demand sensitive to developments in such customer industries and has created difficulties in rewriting code for the newer open systems. It has also made large numbers of existing programmers and software engineers heavily specialized with limited skills or interest in developing more generalized packaged software solutions or learning new programming languages.

9) Unlike business related computer software, where software is an input into the production of goods and services, games are sold directly to the end user and represent a final product. This fact has led to significantly different strategies and economics both for the supplier and the user. In this industry segment, a price based strategy to capture global market share appears to be quite successful. At the same time, this situation has made game software developers and systems manufacturers more vulnerable during periods of technical change as systems move to more powerful microprocessors, i.e. 2K to 4K to 8K to 16K and now to 64K bytes per second. Nevertheless, as newer, more powerful game machines have come to use the same family of microprocessors as more sophisticated computers, it has made sense for game developers to adapt their programs to work on multiple systems and thus expand their user bases. Thus, Japanese game software developers are beginning to pursue aspects of the hub and spoke strategy
evidenced by their US counterparts in packaged business software. The success of Japanese game system and software developers, however, has also attracted much of the entrepreneurial talent and imaginative programming skills that otherwise might have moved into producing packaged business software.

10) Finally, US software developers have been effective in capturing and dominating large segments of the Japanese packaged software market. This has been a logical extension of the benefits of a large global user base. This stiff competition in combination with large clients’ demand for customized software and the success of Japanese game developers has attracted Japanese programming and software engineering resources in three directions: one, developing and maintaining large proprietary systems, two, localizing and adapting foreign packaged software, and three developing and promoting game related software. These powerful trends work against the allocation of resources to the development of other packaged software.

PROJECT ACHIEVEMENTS

The project has completed its examination of the evolution of Japan's competitive strength in the computer-related software industry over time, the experience of Japanese firms in transferring successful corporate practices from hardware to software production, and the impact of competitive interaction between Japanese and US software firms in business and government policy. As noted above, several results have important policy and strategic implications. Based on the research data gathered from the interviews, questionnaires and analysis of other materials, the project’s basic hypothesis appears valid. That is, leading Japanese software producers and most major customers have developed their software systems in an evolutionary manner to incorporate specific technologies and routines to achieve definite business purposes. These systems and routines in turn
largely determine their future software requirements. Because of this evolution, Japan’s software
industry now faces some distinct competitive disadvantages that Japanese policy makers are trying to
address. In addition, the study was able to identify several areas in which accepted industry wisdom
turned out to be wrong, or where Japanese institutions and management practices normally
considered benefits relative to US competitors, in software appeared as weaknesses. Some of these
conclusions have just recently begun to appear in the media and in the statements of industry analysts
as well, giving additional credibility to the research results and the project’s overall analysis.

From the research results, some of the systemic situations facing Japanese firms in this industry,
several of which were unexpected, include the following:

1) Researchers had expected that given Japanese computer manufacturers outstanding technical
advances in hardware that they would be equally advanced and adept in adopting new software
innovations. However, we found that large mainframe producers are confronting substantial
difficulties in modifying their production processes to accommodate rapid changes in software
technology. Their past competitive success in manufacturing and competing for market share
based on distinct operating systems seems to have hindered their adaptation to new
circumstances. Further, many continue to be successful hardware manufacturers, limiting the
resources they can devote to software development. Further, applying their successful
manufacturing routines to software has done little to develop a large user base for packaged
software. Rather, it has helped them control the cost of continuing to develop customized
applications. To maintain technological parity, many have entered into strategic alliances with
US and other foreign software developers to adapt their software to their operating systems.

2) Large development costs and the low cost of reproduction have made user base expansion a
primary goal for software producers. The relative ease of acquisitions in the US relative to Japan
gives US firms a distinct advantage in rapidly building or expanding such a user base. Indeed, responding to such forces, firm expansion via acquisition appears to be a major trend in the US and Canada as witnessed by several large recent transactions across a range of software segments, e.g. IBM-Lotus, Adobe-Aldus, and Computer Associates-Legent.

3) Early conversations with many industry experts led the team to expect rapid technological convergence with the US with respect to downsizing and open systems. Indeed, several have predicted the end of the mainframe and its replacement with network servers as inevitable. MITI surveys regarding user interest also support this conclusion. There were corresponding predictions for a rapid growth in packaged software sales. However, this convergence is not occurring. The increased use of servers and workstations for downsizing while progressing in Japan is only happening at the margin. Further, even what is happening will take 10 to 20 years for major firms to achieve. Even then, mainframe activities will still represent 30% to 50% of large customers EDP expenditures. Downsizing and the shift away from mainframes is thus proceeding only very gradually in many industries and end uses. The degree of shift in turn appears closely related to the need for large data bases, security and access control, existing programming systems (i.e. installed cost), large processing or computational requirements, and high speed mission critical operations. There is no indication that Japanese managers in leading firms are more capable than their US counterparts with respect to software usage or technology. Nor is there evidence they are adapting more quickly to changing technology or market conditions by adopting "best available practice". Indeed, they seem to be moving much more slowly. Reliability and continuous operation appear to be more critical variables than possible cost savings or increased flexibility. The organizational changes required to maximize the use of such systems also seem to be difficult for Japanese firms to adopt with downsizing and potential
increased labor redundancy actually a possible negative influence.

4) At the same time, customized Japanese application software in the areas of Japan’s competitive strength such as autos or consumer electronics production technology is quite competitive on a global basis as a part of Japanese foreign direct investment (FDI) in these industries.

5) Conversely, several US firms, especially in financial services, are using software technology and their ability to rapidly develop various software support systems as a competitive lever in the Japanese market. This represents an important area for future research.

6) While most large users could not name a single major Japanese producer of packaged software other than for word processing, they were unconcerned, since they did not feel dependency on adapted foreign software was a business or policy issue so long as it met their requirements. Further, their continued emphasis on customization and the relatively small 5-15% share (depending on the data source) of packaged software (1991-1994) in overall software use means their dependency has been relatively small.

7) The permanent employment system often considered a plus in early high growth periods is potentially a negative for software developers. This is because software engineers and programmers feel little compulsion to leave their existing firms to start new ventures. This also supports an environment where their skills are highly focused on the requirements of maintaining and expanding a specific company’s existing large installed base and proprietary software system. It thus facilitates the extensive use of customized software and the necessary resource allocation.

8) These institutional arrangements have tended to lock the EDP departments in the large
Japanese firms more into their software history and older programming languages like COBOL than their US counterparts who purchase more packaged software. Under these circumstances, cost improvements come more from experience curve economics and than from the user base economics that determine global competitiveness in software.

9) The researchers were surprised at how readily MITI officials admitted their past policies had failed and that they were no longer actively supporting the TRON and Sigma projects. Rather, they were looking for new policies. A related and important research result has been that unlike the situation for computers and semiconductors in the 1960s and 70s, government has little influence over software as large customers and producers pay scant attention to its policies and initiatives.

10) The project team had initially assumed that the large integrated systems producers were the dominant force shaping the software market just as they have been in hardware. However, during the course of the project, they discovered that influence has shifted to the large systems customers. This shift in turn has implications for Japan’s entry into the Information Age as a “Supra Industrial Society” as opposed to a “Post Industrial Society”. Some analysts have hypothesized that Japan’s transition to the Information Age will be adversely impacted by its weakness in software development. The results of this study indicate that they will just be taking a different path.

11) The research results show the Japanese industry’s weakness is due to continued fragmentation based on customer economics not any cultural advantages the US may have in writing software or US managers’ ability to better or more quickly introduce new software systems. In turn, this fragmentation has perpetuated the overwhelming allocation of Japan’s computer software resources to the mainframe and customized market. There is in fact a cycling
effect at work where the existing base of incompatible operating systems and installed proprietary software necessitates constant customization even of packaged products to both maintain and upgrade each customer’s system. Having put in more resources, the existing commitment is expanded forcing the continued allocation of resources in the future, including the training of personnel to use older programming languages. This situation is accepted due to the small impact on final product cost. But it has facilitated foreign firms’ dominance of packaged software for PCs and workstations given their large global user bases and close relationships with the most widely diffused microprocessors and operating systems.

12) With respect to protection of intellectual property rights, the team discovered through the questionnaire and interview process that in a customized fragmented software environment, such rights are of less concern, since firms have many other ways, including tacit knowledge and lifetime employment, to protect their software. Thus, this is more of a business and policy issue for the foreign firms selling packaged software. At the same time, weak copyright enforcement, makes it very difficult for domestic packaged software producers to develop and survive, especially given stiff foreign competition.

Dissemination of these and other policy oriented research results have taken several forms. Papers were presented at the Association of Japanese Business Studies and the Japanese Economic Seminar at Columbia. The Center on Japanese Economy and Business has published and disseminated a Working Paper entitled “Soft Policies and Hard Competition”, which is appended. In addition, Professor Marie Anchordoguy has presented a related paper at a Prism meeting in Washington. These presentations in turn elicited inputs, comments and suggestions which have been helpful to the research team in preparing this final report. Due to the project, the primary researcher was also able to raise funds for a joint US-Canadian conference to be held in
Victoria in September on: Japan, Computers and Intellectual Property. The primary researcher will be presenting a paper in September at the Canadian Association of Japanese Studies on the strategic business use of software in Japanese competition as well. Japan Access has also agreed to distribute parts of the Final Report to their membership. Finally, the team is incorporating many of the projects conclusions in a simulation of the major factors determining market share in various Japanese software market segments. They hope to make this available to those interested in using the model.

POSSIBLE FUTURE ACTIVITIES

This project has stimulated our interest and knowledge so much that we intend to go beyond the original objectives by engaging in further research in this area, funded from other sources. The project team intends to further discuss its research results and their implications with industry participants and other researchers over the next few months as part of its effort to produce a book on the industry and to experiment further with the simulation. This will include a trip to Japan. Building on the various materials gathered and this report, the team has plans to do additional analysis and comparisons between the Japanese and US responses to the questionnaires and the indicated differences in industry structure and competition to further identify interesting areas of contrast and comparison as they affect global competition not only in software but in industries such as finance which use software technology to compete. European software developers’ responses, however, indicate they are not following a US or Japanese pattern, but are generally seeking out and developing specialized niche markets. Thus, they appear to offer less opportunity for this type of global competitive analysis.

The primary researcher also plans to do additional work with the Japan Users Association for
Information Systems as they have provided the team invaluable assistance in approaching major software users and developers and have offered to assist in reviewing the material for the book.
Final Report of Research Results on Japan’s Software Industry:

SOFT POLICIES AND HARD COMPETITION:

The Impacts of Producers, Users, and Government on the Development of Japan’s Software Industry

Submitted By

William V. Rapp and Hugh T. Patrick

Center on Japanese Economy and Business
Columbia University
New York, NY 10027

212-854-3496 (Fax: 868-3107)
TABLE OF CONTENTS

Introduction and Preview

Executive Summary and Research Approach

History of Japanese Computer Software Industry

Types of Software

Economics of Software
  Producer Economics - Experience Curves and User Bases
  Customer Economics - Customization and Installed Systems
  Economics of Packaged Software, Including Games

Hardware, Technology, and Market Segments

Industry Structure, Organization and Investment
  Mainframes
    Office Computers, Minis and Network Servers, Workstations, & PCs
  Game Systems

Government Policies, Objectives and Initiatives

Competitive Dynamics
  Evolutionary Approach
  Alternative Paradigms

Comparisons and Contrasts with the US

Analytic Results

Future Evolution
Summary and Conclusions

References and Bibliography
TABLES and APPENDICES

TABLES

Table 1: Estimated Japanese Software Market Size by Segment 1992

Table 2: Results JISA Survey of Large Japanese Customer Software Market: Relative Size by Market Segment and Growth Rates

Table 3: Growth Rates for Japanese Software by Market Segment 1990-94

APPENDICES

APPENDIX I - Simulation Analysis and Explanation of Simulation Variables

APPENDIX II - English Version Japanese Questionnaire

  for Integrated System Producers

  for Software Developers

  for Large Users

APPENDIX III - Japanese Version Japanese Questionnaire

  for Integrated System Producers

  for Software Developers

  for Large Users

APPENDIX IV - US Questionnaire for Integrated Systems Producers

  US Questionnaire for Software Developers

  US Questionnaire for Large Users
APPENDIX V - Questionnaire sent to European Software Developers

APPENDIX VI - List of Cooperating Firms, Groups and Institutions

INTRODUCTION AND PREVIEW

This report presents the findings on the Japanese Software Industry of a research team led by Professors William Rapp and Hugh Patrick under the auspices of the Center on Japanese Economy and Business at Columbia University funded primarily by a grant from the Japan-US Friendship Commission with additional support by the CIBER at the University of Washington, Columbia University and the University of Victoria. The team explored policy and competitive aspects of the Japanese software industry and its development primarily from an historical, structural and strategic perspective. Based substantially on the evolutionary theory of enterprise organization and competitiveness, the project team developed an integrated, logical explanation of the reasons for the failure of MITI’s software industry support policies, Japanese success in games, the persistence of large scale customization, the development of group based software subsidiaries, and several other important phenomena unique to the Japanese market. In addition, the team can demonstrate why these industry characteristics are likely to persist and resist global convergence even in the face of powerful technical forces that many argue will force the Japanese software market to conform with the rest of the world and particularly the United States.

In turn, the likely continuation of these trends convinced the researchers that except for games, Japanese firms are unlikely within the next several years to become major players in the global computer software market. This conclusion early in the study shifted the research focus from investigating the question how by building on a Japanese base Japanese firms might enter and compete strategically in the Asian and North American software markets as they had done in
steel, autos and semiconductors. Rather, the more important research question from both a policy and strategic point of view is to understand why US firms have been competing so effectively in the Japanese market and what has been preventing or blocking the development of a globally competitive Japanese software industry despite MITI’s efforts to promote one. This would then permit the researchers to assess whether these factors were relatively permanent or would be modified by changes in policies or corporate practice. This shift in research focus in turn meant the team put its primary research emphasis on understanding the competitive dynamics of the Japanese software market and in discovering foreign software firms’ competitive strengths and how they were utilized in that market.

The analysis and comparison with US and European software firms was then used to identify and confirm the source and institutional underpinnings for foreign firms’ competitive strengths to determine if they could be easily replicated by Japan and potential Japanese competitors, or were inherent to their industries and the way their firms had developed. The project thus focused on the Japanese case in depth, rather than trying to duplicate it at a less sophisticated level for the US and Europe. That was not necessary to answer the research questions posed concerning how the Japanese market for computer software was developing and the strategic role in that development of foreign competitors, the bulk of whom were US, with a scattering of European, Canadian, and Australian companies.

The team closely examined the hypotheses that the failure of Japanese users to quickly adopt best practice in terms of new software technology and systems was due to bad management or inadequate understanding and appreciation of new technologies. They similarly assessed the
conclusion posed by some analysts that Japanese are at a cultural disadvantage in writing or developing software (Delaney 1994) due to language and education. Both explanations were found to be unsatisfactory because they could not explain major discrepancies in predicted outcomes. They were not satisfactory explanations of Japan’s difficulties in creating a globally competitive computer software industry. User responses showed that they knew about and were adopting new software systems and technologies but were proceeding very slowly due to their existing installed bases and their conservative, risk adverse corporate cultures. When given an opportunity to switch after the Kobe earthquake, when several large firms’ computer systems had been destroyed, those companies moved aggressively to adopt newer network solutions rather than try to replicate their old systems. Similarly, the argument that Japanese find programming difficult due to language or education is inconsistent with their global dominance in game software.

Rather, as presented in the Executive Summary and in the historical section and the section on industry economics, the research results indicate that Japan’s problems building a competitive computer software industry, other than in games is, due to users’ strong and rational preference for customized software even though it is more costly, while in games this preference set does not exist. Thus, an evolutionary analysis of the industry and its competitive structure was found to be the appropriate research approach and was the one pursued throughout the study as other possibilities were discarded as lacking in explanatory power. The continued strong demand for customization is a critical element in this assessment.

This is because many industry experts have expected the demand for customization to
fade along with technical convergence and a long-term trend towards open systems and downsizing. However, customization has persisted and actually grown, in most cases faster than its packaged counterpart. The reason for this is that customization over the years has interacted with the large variation in Japan in computer operating systems. As discussed in the historical section, the current multiplicity of operating systems is due to decisions made in the 1960s regarding foreign licensing arrangements of computer technology. However, this multiplicity when combined with various users’ proprietary systems serves to fractionalize the software industry and prevent the development of standard packaged solutions, frustrating MITI’s computer software industry development policies. This explanation is utilized in the section on government policy that describes how and why an independent world class industry has not developed in Japan even though MITI has followed their traditional industry support model.

The sections on large customers and systems suppliers’ economics and software preferences show that foreign (mainly American) software producers with rapidly growing global demand for their new products that support network servers, workstations, and PCs have successfully pursued a hub and spoke strategy to expand their Japanese market share. In such a strategy, the foreign software developers localize and adapt their existing software packages to the multiple operating systems that exist in the Japanese market. Their success in doing this has then expanded their global user bases and improved their costs, putting local Japanese software developers at an even greater disadvantage. The research results predict this situation will continue, favoring intra-industry software advances by major users and their captive systems developers over inter-industry expansion. A strong yen also contributes to the competitive
strength of this process by making Japanese packaged software less competitive domestically and
globally, forcing resources back to the customized market where there is continued and growing
demand.

Thus, the research leads to the conclusion that the Japanese software industry’s strength lies predominantly in customizing software to meet client demands. Accordingly, the Japanese software industry is at a competitive disadvantage in non-mainframe and packaged software, and faces strong foreign competition with respect to the latter even in the Japanese market. However, this does not significantly reduce the competitive strength of manufacturing and service firms that are software customers. They have developed corporate strategies which incorporate software as a significant and specialized input to their production systems, but in ways which do not require dramatic alterations in those production systems.

The report begins by presenting in the Executive Summary a synopsis of the overall approach to the basic research goal of explaining the current competitive structure of Japan’s software industry and how it is likely to change in the future. In particular, this section explores whether the current market advantage of foreign producers is likely to persist and whether the emergence of a major Japanese software competitor seems probable. It explains that, in addition to the increase over time in foreign firms’ appreciation of Japan as an important market and Japanese firms as potential competitors, the software industry has several unique characteristics that separate it from past Japanese successes such as steel, autos or semiconductors. It also discusses the comprehensive research approach used by the team to identify such characteristics and to answer questions concerning its future evolution. The research included data collection,
review of other studies and analyses, meetings with industry experts and government officials, interviews with large corporate customers, systems producers, and software developers, the distribution of questionnaires and a simple computer simulation.

The report then establishes the context for understanding the Japanese computer software industry by charting its early history and particularly the origins of its multiple operating systems in the section providing a brief History of the Japanese Computer Software Industry. Important features of the Japanese computer and computer software industry are emphasized in this discussion of its early development in the 1950s and 60s, such as MITI’s policies for encouraging alliances and licensing arrangements between large foreign producers and Japanese firms interested in entering the computer business. Unfortunately, MITI failed to realize that as a high growth industry, computers did not yet have either a stable competitive structure or an established global standard as was the case in automobiles, steel or DRAMs. Thus, several large Japanese companies lost their outside technical support as their affiliated foreign firms withdrew from the market. At the same time, stiff competition from IBM stimulated the Japanese computer firms (integrated system producers) to maintain these operating differences as a way to lock their customers into their own proprietary systems. They also gave away software as a way to sell hardware, completing their encirclement of their customers. This was in turn followed by a series of industry and government actions related to advancing computers and software such as the fifth generation computer project, the TRON project and the Sigma project, which have failed to provide any real progress or integration but have rather brought us to the present day with many of the original operating legacies still intact.
Due to the great progress in producing smaller and more powerful integrated circuits, the computer industry has produced smaller and more powerful computers. This has led to the development of new types of computers and in turn new computer software to run them. This represents a large source of market growth and has deeply influenced software’s technical evolution. The next section on the Types of Software thus explores the impact of this technical evolution on the different software operating systems used by different computer hardware systems in Japan. This section also explains the differences among various types of software and their use: operating systems that run computers, middleware that incorporates various industry protocols and the standard communication links between them, and application programs that perform user specific tasks. This section is designed to help those who are not industry specialists understand the differences among various types of software and the market for each. It also shows how certain software is directly linked to the growth in the demand for certain hardware and functional client requirements.

While differences among software programs and operating systems affect competition within and between market segments, there are common factors influencing a firm’s ability to compete in the market that are common to all types of software. In the section on the Economics of Software, these attributes are first analyzed generally and then with respect to the Japanese market specifically. Packaged software producers spend a great deal of time and money developing specific software programs that can be easily reproduced and distributed. This is an attribute that they share with the pharmaceutical industry. For these reasons, they are also dependent on strong intellectual property protection to foster their development since otherwise
they cannot insure a steady cash flow for upgrades and future projects. In addition, large
development costs mean that the size of the potential and existing client user base has a greater
impact on an firm’s cost structure than improved programming skills. The user base can also
create external benefits to users that need to share information within a client or external to the
firm. Conversely, customized software serves the needs only of a specific user or set of users
within client and thus is protected by being imbedded within their overall corporate Electronic
Data Processing (EDP) system. The competitive and strategic implications of these and other
factors on Japanese producers is explored in the first part of this section.

At the same time, software producers face many different constraints and economic issues
than those their customers face. This section therefore also compares producer and customer
economics: their commonalities, differences and interactions. As explained in the section on the
industry’s history, initially producer economics and initiatives governed the development of the
computer market and in turn software. MITI’s classic producer oriented development strategy
worked for computer hardware development, and since software was given away as part of a
bundled transaction, it followed along. However, this paradigm did not work for developing a
software industry when producers’ interest in giving away software was limited to that which
would sell their hardware, i.e. software only usable on their proprietary system and those of their
customers. This structure did not support the development of the large common user base
necessary to be competitive in packaged software. That is, it was difficult to develop the
economies of scale for standard software packages such as they could for example in
semiconductors.
Further, as customers developed large installed bases and their own software systems and infrastructure, their own economic criteria changed. Compatibility, communication and interoperability among diverse operating systems and hardware/software configurations became important to them as different functions and uses were best met by using systems other than mainframes. Yet, these installations needed to be able to get and share information. In the second part of this section, therefore, software usage and economics are examined from a customer’s viewpoint. Here, based on the questionnaires and industry interviews, the potential economic benefits of customization are examined and are set against the cost benefits of the large user base and externalities available through using packaged software. This section also identifies the economic importance to large customers of their existing installed systems and the intellectual property protection that customization can provide for the corporate skills incorporated in their proprietary software systems.

The section on Hardware, Technology and Market Segments which follows considers the significant influence of differences in computer technology, hardware and computing power on software’s use and the software industry’s structure. It traces the evolution through time of smaller and more powerful computers such as minis, then workstations, and finally PCs, and the corresponding changes in the market for that hardware and its corresponding software. More particularly, the historical development of specific hardware/software configurations, often pioneered by different manufacturers, has meant the development of quite distinct market segments based on these differences, with some competitive overlap. This section identifies the following areas as providing the biggest separation between operating and hardware systems in
terms of the competitive dynamics of a specific market segment: supercomputers; mainframes; mini or office computers often used as network servers; workstations; PCs; and game systems. The study in turn focused its analysis on mainframes, network servers, workstations and PCs as being the primary areas of interest since they represent the interest of the large customers that are a central focus of the study. At the same time, the size and volume of the computer game market is such that a comprehensive explanation of the business software market should also clarify the dynamics of the game market. This section raises the issue that, due to the trend among game machine producers to use the same family of microprocessors as major computer manufacturers along with the emerging hub and spoke strategy of several game software developers, greater structural interaction between the software sectors of games and business use is emerging.

Each of these hardware/software categories has its own technology, competitive and economic forces at work. The global computer industry trend towards smaller more powerful systems does mean there is some competitive overlap between computer uses and their market segments. Yet, on the whole each has different competitors and competitive characteristics within the Japanese market. This technology trend toward more computing power and greater software capability in each segment in turn affects the usage of all computers, acting to dynamically segment the market relative to mainframes, network servers, workstations and PCs. The section particularly examines the evolution of the hierarchical “3 tier system” as one aspect of this effect in the Japanese market, which has mainframes managing network servers that manage workstations and PCs. It also analyses the relationship of these technologies and market segments to each other from a system producer’s, software developer’s and customer’s
perspective. So whereas integrated systems producers will try to optimize vertically through a family of machines and microprocessors (MPUs), customers are interested in horizontal compatibility and the contribution to their total system. Software developers will take their cue from whether they are more closely affiliated with customers or with the large integrated systems producers.

The next section on Industry Structure, Organization and Investment separates each of these areas defined by a particular computer technology into software market segments, each with its own special competitive and growth characteristics. The growth and type of software sold is different for each market segment: mainframes, network servers, work stations, PCs, and games. The mainframe market is dominated by the established integrated systems producers such as Fujitsu, NEC and IBM Japan. Hewlett-Packard and Sun are the leading players in workstations. Appreciating the way this situation has evolved and the competitive strategies of the different players is very important to understanding the competitive dynamics and evolving structure of Japan’s software markets and industry. In particular, this section extends the concept of a hub and spoke strategy in which software developers adapt their software to different operating systems in order to increase their user base and improve their competitiveness, which then allows them to pursue additional market share.

The role of government policies in this dynamically evolving environment is addressed in the next section on Government Policies, Objectives and Initiatives. This section outlines and examines the various policies that the government has pursued in trying to develop a world class software industry comparable to steel, autos or semiconductors, and considers why these policies
have failed and been abandoned. It also assesses why, even though software is a targeted industry, no Japanese company users interviewed directly or by questionnaire is taking the government’s plans or policies into account in determining their own EDP or strategic development plans. Further, MITI’s traditional producer oriented policies seem to have less relevance in the new customer dominated market. The first mover advantages in this industry are powerful so that a fast follower strategy appears less appropriate as a way to develop and grow. At the same time, MITI’s past policy successes in other industries are seen as moving the software industry in a different direction, since the auto, steel and consumer electronics industry have established their own software subsidiaries and are continuing to compete successfully in their global markets through constant improvement in their production processes, including their software systems. Their exports and foreign direct investment thus contain this advanced customized software. This section thus addresses the question of whether the main source of MITI’s current difficulties in the development of a packaged software industry are to be found in its past success in other manufacturing industries.

The historical context, technological trends, customer and producer economics, industry structure and government polices combine to provide the Competitive Dynamics, the subject of the next section. These dynamics are what uniquely characterize the Japanese market for software and its longer term development. The market is segmented along several lines in terms of hardware and its associated software, packaged versus customized, and operating systems versus middleware and application programs. Each type of major hardware category -- Mainframes, PCs, etc. -- has different operating systems. For example, while most Japanese
mainframes are IBM compatible in that they have the same IBM instruction set wired into their hardware and were designed to run their customers’ existing IBM application programs, each has a somewhat different operating system so that applications developed for them specifically will not run on a competitor’s mainframe. In turn, MAC and IBM compatible PCs have different operating systems both from each other and from that used for running an IBM mainframe. Between the operating systems and the Application Program Interface (API) lies a range of middleware that includes the communications programs used to run networks. While operating systems are generally standard or packaged for each hardware supplier, middleware can be either packaged or customized depending on the customers’ requirements for a particular function. The largest software market is for applications and represents over 80% of the market. These in turn can be prepackaged software, such as word processing and spread sheet programs, or highly customized requirements such as derivatives trading or managing a factory’s just-in-time manufacturing system. The language programs used to develop these application programs such as FORTRAN, COBOL or C++ and their associated compiler programs are generally referred to as tools and represent another software market segment. While the market for middleware, tools, and application programs are also segmented, the specific dynamics of these subsegments go beyond the scope of this study. However, they are affected by the same user base economics, institutional influences, and hardware technology that impacts the larger market categories.

The structure, forces and trends in the Japanese software industry are then compared with the US industry in the section on US Comparisons along four dimensions specified by Edward Steinmueller in his study of the evolution of the US software industry. Comparisons are made
with the US based on the interviews and questionnaires with US companies. The results indicate that the Japanese industry is much more wedded to customization than their US counterparts and that unlike Japan the strong development of the captive software subsidiary of the large customer does not exist in the US. These factors in turn produce different views towards reengineering and less demand for downsizing and its associated hardware in Japan.

The next two sections discuss the Analytic Results of the research and the probable Future Evolution of the Japanese software industry given these results. These findings and the team’s conclusions are then presented in the Summary and Conclusions. The Analytic Results section explains in more detail the information gathered during the study from various sources, including the interviews and questionnaires. Each Japanese customer is seen as pursuing its own EDP goals, with government having little impact on the process. This pattern is seen as persisting so that the software industry’s future evolution will be heavily determined by customers’ continued emphasis on customization and semi-customization carried out through captive software subsidiaries. At the same time, American and other foreign producers will pursue a successful hub and spoke strategy in selling in Japan that permits them to effectively access this type of market through cooperation with the large integrated systems producers such as Fujitsu, NEC and IBM Japan. Such a strategy improves their cost economics and competitiveness by increasing both their global and local user bases, thereby reducing the unit cost of the basic program and the cost of localization for the Japanese market. This makes it more difficult for a Japanese packaged software producer to develop a competitive program and to compete against foreign software suppliers, driving them towards game programs where the same sort of
beneficial economic cycle is possible for them due to the large, unfragmented, consumer market for games in Japan and the rest of the world.

The Appendices provide the documentation of the research procedures for this project. It begins with a discussion of the simulation methodology which captures in mathematical and computer software form for the Japanese packaged business software market the basic relationships and variables specified elsewhere in the report and particularly those presented in the sections on analytic results and the industry’s future evolution. Subsequent appendices provide the various questionnaires, in English and Japanese, used to obtain information from Japanese and American firms that are integrated systems producers, software developers, or large customer users. There is also a questionnaire used to contact European software producers. This is followed a list of most of the many firms, groups and institutions who cooperated with the study. Also appended is a earlier manuscript generated from this study, presented at various academic fora and sent to industry respondents, as a basis for peer group feedback.
EXECUTIVE SUMMARY AND RESEARCH APPROACH

This report’s title “Soft Policies and Hard Competition” refers to the apparent role reversal in the Japanese software industry between Japanese government and industry policies on the one hand and the aggressive pursuit of market share by foreign competitors on the other. Historically, Japanese government policy in targeted industries such as autos, computers and semiconductors has been hard-nosed and quite effective in promoting the development of global Japanese competitors who have competed on price and quality in pursuit of greater market share (Abegglen and Stalk 1985). At the same time, many analysts have criticized foreign firms’ for failing to really compete on the basis of price, quality and direct investment in Japan, thus giving potential Japanese competitors time to develop (Rapp 1977 and Abegglen and Stalk 1985). Yet, in the software industry, Japanese government policies have generally been weak and ineffective, while foreign firms have established an increasingly strong local presence and have successfully exploited the market’s fractionalized structure, including in recent years gaining support from large Japanese computer manufacturers. This is despite the fact that the major factors determining this trend rest in the decision-making processes of MITI’s natural constituency, namely large Japanese firms, both software producers and customers. The report also explains why this trend means the Japanese software market will not move towards the general adoption of packaged solutions and technical convergence with the US, despite the predictions of many analysts.

This report presents the research results explaining this role reversal through an analysis of the development of Japan’s software industry from an evolutionary perspective. The
evolutionary approach proved 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 catch-up development model. Based on this analysis, the research team then assessed the industry’s current strategic structure and has forecasted its likely evolution in light of the key business and economic variables affecting the industry. These include the relative administrative heritages, roles and motivations of large customers and suppliers, as well as government policy makers, as they shape decisions regarding software usage and thus the market.

The Japanese software industry is part of Japan’s Information Services Industry which also includes online services, VANs (Value Added Networks), and other processing services. In 1990 total information service revenues were 7 trillion yen having risen 26% from 1989. However, since then they have remained relatively flat (MITI 1993 and Boyd 1995) primarily due to the effects of the collapsing bubble economy. Sales by the software industry in turn have accounted for about 60% of this market or about 4.2 trillion yen (MITI 1993). Historically and currently the software industry is characterized by a strong large client preference for customized software over packaged products, especially by large customers. Large manufacturing and service companies have developed highly customized software systems that reflect their specific organizational structures and operating advantages. In the past, this was encouraged by the large integrated systems producers such as Fujitsu, NEC and Hitachi. Because of the foreign alliances these producers entered in the early 1960s, and their weak competitive position relative to IBM,
each had a different operating system even if they were using and licensing IBM’s instruction set for incorporation into their hardware. This was also true for foreign mainframe and mini computer producers such as IBM, Sperry-Rand, Burroughs, DEC, and Hewlett-Packard. Therefore, by giving away bundled software to sell hardware and by helping large customers develop their own proprietary software system solutions, they could lock a customer into their own incompatible systems. As such, the structure of the Japanese software industry today reflects this administrative heritage of the major customers as much as developments in computer technology.

Since most large customers in turn developed their competitive positions by importing foreign manufacturing and other technologies and then improving on them, large Japanese computer customers tend to emphasize process over product innovation in their competitive strategies (Baba 1989 and Rapp 1993A). Because software is viewed as an input into the production or service delivery process rather than an output to be independently marketed, from the large customers’ standpoint their unique, proprietary software systems are a critical part of what makes them competitive. From their perspective, software as a production input does not show up as unit sale of packaged software; rather, their software is sold as one cost component of their finished product or service (Dalton and Genther 1991). However, the significance of this perspective on the development of Japan’s software industry has only recently become apparent.

This is because major Japanese customers have now established their own software companies to provide software to them and their keiretsu related companies. Five of the top ten software/information service companies are so affiliated. NTT Data, Nomura Research Institute,
and Enicom (Nippon Steel) are examples of such companies. Two others are related to Hitachi. This corporate structure is seen by the parent companies as a way to control software costs, to build upon their existing specific customized software systems, and to provide an independent career path for their EDP professionals. At the same time, these affiliated software firms are committed to maintaining their parent’s and their group’s existing software, systems that can run into hundreds of millions of lines of code. This entails the commitment of significant computer, programming and financial resources. Some but not all of these captive firms also solicit outside business up to a limit of about 50% to partially defray the costs of maintaining their group’s proprietary systems. In these cases, they are competing with the software subsidiaries of the major integrated systems producers and some of the independent systems integrators, though apparently many of the later are really independent in name only and rely heavily on a particular client or industry for their work. Since this structure is designed to maintain these large customers’ unique proprietary systems and competitive prowess in industries such as steel, autos, consumer electronics, electronic components and financial services, it perpetuates the large market for customized software. However, the continued high demand for customization combines with large variation in operating systems to fractionalize the industry and frustrate government support policies for a large packaged software industry. These firms also seem to have been increasing their share of the overall software market as more large firms form such subsidiaries and smaller independent companies have gone bankrupt due to the recession.

Industries in which Japan has shown particular competitive success are key players 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]), and 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. In this sense, software is customer-driven and becomes an integral part of a firm’s global market advantage.

Nevertheless, this demand for customization has not adversely affected the foreign suppliers of packaged software. Indeed, it may have helped them. This is because as Japanese customers have built their own systems and expertise, they have become relatively more powerful than the large systems producers, and the market has shifted from one primarily influenced by producers to one more influenced by the customers. In this situation, as reflected in recent MITI surveys (JIPDEC 1993), customers are interested in reducing their dependence on one supplier and moving to a multiple vendor system. They also wish to access newer more open, flexible systems and technologies such as network servers, work stations and PCs (JIPDEC 1993) provided that they can be integrated into their own proprietary systems. This has led to rapid growth in semi-customized solutions where packaged software is substantially customized to a specific user’s requirements. From the questionnaires, it was discovered that the cost of the standard software package will typically run only about 30% of the total cost of such a semi-customized solution as compared 70% to 80% in the US. While this may seem like a heavy
penalty for Japanese customers to pay, most seem to feel they recoup this cost in the improved overall efficiency and productivity of their operations within which software is a key component.

American and other foreign packaged software developers in turn have been effective in penetrating this market through the development of a hub and spoke strategy whereby they adapt their software to the operating systems of the major systems producers, who in turn help them market the product to their large clients who then customize it. Most major systems producers, including IBM Japan, have offices in the US to facilitate this process. The systems producers benefit in that they continue to keep their major clients’ software technology current, and to control the relationship. The customers are happy because they keep their systems technologically current at lower cost while freeing resources to do pure customization or more semi-customization. The foreign producers are pleased as they expand their global user bases as well as further spread the cost of localizing their product for Japan, a requirement that usually doubles the cost per unit sold.

User base economics plays a large role in these outcomes. Software, like pharmaceuticals, has very high development costs and the time to develop a new program can be quite long; the cost of reproduction, packaging and distribution, though, are quite low. This means that unit costs are extremely sensitive to the size of one’s user base. In the case of customized software, the user base is one -- there is only one sold to one user within a large customer. As the software supplier obtains multiple users, the costs per unit sold drop dramatically. The more expensive and complex the program, the more impact a larger user base can have in determining competitive advantage. Selling several copies to one large customer can be a very effective way
to do this. This explains the large customer’s importance to the PC packaged software supplier and the PC packaged software market. The typical success story of a US software firm is therefore one that develops a new product and quickly achieves market dominance, often 70% or more. In turn once a user has learned how to use a program and has installed it as part of an overall corporate system, the user and the customer will be very reluctant to switch. This creates a barrier to competitive entry. While this barrier can be overcome, as in the case of Microsoft’s success with its Japanese Word Processing program, Word for Windows 3.1J, it is usually done from a related user base and in a high growth market where new users can be added on the basis of price rather than by trying to take existing users away from a competitor. For this reason, US firms are constantly endeavoring to expand their user base on a global basis, using their cost advantage from the global market to enter each successive local market and then improving this advantage by capturing a large share of that market. Japan, as the second largest software market in the world, has become an important locus for implementing this strategy. Localization costs drop too in response to an increased share in that local market. Since despite its size the Japanese market is so fractionalized, Japanese software developers are at an inherent disadvantage relative to their foreign competitors with access to global markets, and large customers have little interest in subsidizing their development by paying a higher price for an untested program with a small user base. Further, as most software innovations take place outside Japan, most foreign competitors already have a large user base before ever entering the Japanese market, while the cost of localization is much less than the cost of new development. This pushes Japanese software development resources towards the captive software houses and customization that does
not compete with foreign packaged software, and indeed is complementary in its semi-
customization form. A strong Yen contributes to this competitive situation by making semi-
customization an attractive option, but it does not appear to be the overriding criteria since
customization flourished during the weak Yen environment.

A major research conclusion is that a Japanese software market characterized by
customized software and foreign software that has been adapted and customized by
manufacturers is a trend that will persist for some time. In this market, Japanese developers will
supply to customized market while foreign firms will supply the packaged market. Penetrating
the customized part of the semi-customized market will vary, depending on whether knowledge
of the package or of the proprietary system is most important. Generally, foreign producers of
package software, though, seem only to want to involve themselves with customization to the
extent necessary to sell their packaged systems. From this view, there seems to be a natural
division between the two markets and Japanese and foreign competitors.

Similarly, because large Japanese customers’ existing computer systems are heavily
mainframe based, the demand for mainframe software, particularly customized software, will
continue to represent the bulk of large customers’ software expenditures for some time. Most
firms interviewed felt that even after ten years mainframe related software will still represent
30% to 50% of their software expenditures. In addition, in incorporating network servers,
workstations and PCs into their proprietary software systems, most large users have opted for a
“Three Tier” system in which one or two mainframes manage a series of network servers that in
turn manage several workstations and PCs. This pattern is also expected to persist; the
mainframes will act as the host computer, smaller mini-computers will act as network servers since that is where the communications program software will be lodged, and workstations and PCs will act as smart terminals which can both operate their own programs and communicate with other smart terminals and the host computer.

The Ministry of International Trade and Industry (MITI) is trying to deal with these realities by shifting the nature of government support for Japanese software developers (excluding game developers) away from large systems producers as represented in the Fifth Generation, TRON, and Sigma projects. Their hope is to promote, with the US as a model, small and medium size software companies whose products seem to be in demand. However, the Ministry apparently still is treating software as an output rather than primarily as an input into the production activities of other industries. In this sense, there is little change from the MITI software strategy that failed in the late 60s and early 70s. Given the existing and evolving industry structure and the nature of the forces determining demand, a better strategy would seem to be to go with the economic and competitive flow by promoting industry enhancing software that will assist major Japanese customer companies to maintain their global competitiveness. Otherwise, the government’s options appear quite limited.

These user and competitive trends are quite powerful since major customers are very 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 potential glitch in their complex integrated software system will result from the introduction of a new type of software system and will disrupt their basic operations at great cost.
At the same time, there is little perceived benefit to such a switch. Such conservatism is of course consistent with their risk adverse behavior in general (Rapp 1994). This concern with the integrity of the business system, combined with better economics, provides an inherent advantage to US software producers who are localizing and adapting an existing piece of software over Japanese software producers trying to win Japanese corporations to new products. In turn, because large customers, large systems producers and foreign firms are focusing their efforts on localization, adaptation, and customization, the Japanese software industry is not using its available programming and software engineering resources to develop packaged software to sell to the global market, except for games. There is no apparent change on the horizon to alter this environment.

Neither does it appear the globalization of Japanese corporations and their possible need for global systems integration will affect their continued demand in Japan for customized software. While foreign staff in subsidiaries abroad may insist on the purchase of certain prepackaged software to work in their own markets, operations in Japan are unaffected, neither controlling nor being influenced by the decision. The large software customers interviewed have not globally integrated their systems, and few have plans to move in that direction. Further, Japanese staff are trained primarily in the computer language used historically by their employer, limiting familiarity with a range of international computer languages and hardware platforms. Many Japanese EDP managers are not computer specialists; instead, they are corporate managers who believe the firm’s software development should stay focused on maintaining firm competitiveness in their particular business. This tendency means Japanese software developers
are primarily organized along industry lines, a pattern that generally applies to independent developers as well (Coultas 1994).

There is also sometimes cross-shareholding between software 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 the keiretsu suppliers of major corporate customers. That is, firms in vertical keiretsu (suppliers and assemblers) are not linked through software in a common integrated system. However, as described above, supplier firms will usually have their programs developed by the software development company of their keiretsu’s head firm, so that 50% or more of the sales of a major customer’s software supplier may be to companies in the same group as the customer, including those in which the customer owns shares.

Vertical integration and group sales may not be a great benefit to the software developer, though. This is because they do not spread the fixed development costs over a wide enough user base. Thus, 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. After localizing their software and adopting it to a Japanese system supplier’s system and platform, they let the system houses or large clients handle the customization to their specific requirements. In this way, foreign software suppliers develop a series of plug compatible software modules that fit into the customers' existing systems, without
threatening either a system’s overall viability or major system producers’ relationships with their clients.

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 is changing as low cost US PCs using localized Japanese software capture more of the home market (AEA 1992 and Choy 1994). The entry of Dell and Compaq is therefore clearly beneficial for Microsoft and Lotus in the longer term, while the popularity of US application programs and other software products helps Dell and Compaq by providing a large existing library of relatively inexpensive programs that users can immediately run on their machines (Choy 1994).

Still, for the time being the main market continues to be 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 that were available to the government at the time it targeted semiconductors for development, and what it faces in software today. Further, it is not clear such tools would even be beneficial. 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 the computer business due to intense
competition from IBM in the global market, Japanese licensees and their customers were technically isolated. They also had to adapt to software developments abroad on their own since their normal sources of innovation had evaporated.

Since IBM was forced to unbundle its software, the global software market has evolved rapidly. As long as software continues to advance rapidly, US producers seem to have 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 charged to current costs. Further, as noted above, Japanese customers have highly integrated systems and are quite risk averse with respect to the software they use. They avoid experimentation, and emphasize software that is totally debugged and reliable (AEA 1992). They see little benefit in pushing the development frontiers and then risk having their production system crash. Based on these research results, this report concludes that the presently fragmented operating systems and software application patterns in Japan will continue. Standardization is unlikely, except for some work stations. Further, since software is a medium to highly skilled labor intensive industry, and Japan not only is a high-wage country but has a shortage of software system engineers, development of non-customized software in Japan is relatively expensive and, especially 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 costs. Unless this preference for customization changes, which appears unlikely, internationally competitive, locally developed packaged software is unlikely to develop outside
of the game sector. Also, there is really no economic or management incentive for users or integrated systems producers to end such customization, given customers’ huge sunk costs and strong desire to support their unique competitive operating advantages in their basic businesses, which after all are not software; and systems producers desire to maintain their customers, particularly for mainframes. This means only games, niche software such as Japanese language word processing programs, and customized software incorporated in manufacturing systems such as steel or automobiles will be competitive.

The difficulties facing the development of the Japanese software industry have caused some observers to wonder if Japan can successfully enter the information age (Coultas 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, the results of this research show that such concerns emphasize inter-industry developments too heavily, i.e. developing a packaged software industry, while slighting intra-industry development such as further advances in new steel or auto engine technologies and new customized software to support them. They also focus excessively on trends in the US. 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 represents an alternative way for Japan to participate in the information age.

**HISTORY OF JAPANESE COMPUTER SOFTWARE INDUSTRY**
How did this divergent Japanese 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 such factors 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 historically 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 customers. 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, as well as Japan’s global success in standardized game software. An explanatory hypothesis must also rationalize the existence of multiple platforms, the strong support for the localization and adaptation of foreign software, the push for allowing decompilation, the shifts in customer preferences toward flexibility and open systems combined with customization, and customers’ increased desire for compatibility between their software needs and their basic
management goals. Only an evolutionary approach based on an examination of the historical origins of this industry and the motivations of the major participants addresses all these issues.

The key competitive factors affecting large firms are their administrative heritage, their process and systems integration, and continuous introduction of 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. Also, even the IBM compatible producers only hardwired the IBM instruction set into their mainframe computers, for which they paid IBM a royalty. Their operating systems, while able to run potential customers’ existing IBM programs, were enough different that application programs written for their operating systems would not run on the operating systems of other IBM compatible competitors. 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 except to the extent of facilitating an initial transfer to their proprietary system (Anchordoguy 1989; Baba et al 1993; Cottrell 1993; Rapp 1993A; and Coultas 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 US 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.” In addition, the government was heavily subsidizing the computer industry’s development and pressured them to buy Japanese systems rather than IBM through their control of foreign exchange licenses (Anchordoguy 1988 and 1989). 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 NEC’s reluctance until recently to adopt a fully compatible MS-DOS PC operating system. Indeed, it was only pressure from Compaq and other foreign PC suppliers that forced even IBM compatible PC producers such as Toshiba to make their preinstalled DOS fully compatible; before that change in firm strategy, anyone wanting a more flexible system would have to purchase DOS/V and load it themselves. Thus, historical ties and various producers lock-in strategies have left Japan with an installed computer base of multiple systems and incompatible platforms, particularly for mainframes, which are strategically difficult and expensive for large customers to change (AEA July 1992). This conclusion is also strongly supported by responses to the questionnaires distributed to large customers, integrated suppliers, and independent software developers, and is further supported by the results of the interviews.

The reason it is now so difficult for customers 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 customers have had their programmers write 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++. 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 and programmers from the more important task of developing programs for new business requirements or maintaining existing ones. Since the existing programs continue to accomplish their basic business goals, large customers 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 from not shifting.

There are several other reasons for the slow pace of customer 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 and a large user base for that software in the US made it pay for US packaged software developers to convert their programs to work with the newer programming languages and operating systems; however, due to the emphasis on customization in Japan, the packaged software market is small (table below and Boyd 1995), and no one user is large enough to justify developing such a conversion 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 relatively
few Japanese 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 shift totally and rapidly 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. Therefore, the best approach in the current adverse economic climate becomes for everyone to shift at the margin. In most industries, with the exception of finance, there is little concern about possible competitive pressures from foreign firms based on superior integrated software systems. The focus is on their Japanese counterparts.

This marginal-shift approach has occurred despite government efforts to promote a greater switch to newer software technologies and to develop a strong packaged software industry. After several false starts in the 1960s and 70s (Anchordoguy 1989), MITI made a strong effort in the early 1980s 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 (5th Generation Computer Project), Sigma and TRON.

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, shipbuilding, 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, with export a logical extension (Rapp 1994).

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 Feigenbaum (1983) and Cusumano (1991, 1992). The prospect (or fear) of such an outcome captured the imagination of many US and European policy makers and executives (Anchordoguy 1989). However, 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. Coultas (1994) reaches the same conclusion based on similar interviews. One reason for this may be that the fast follower concept is heavily based on experience curve economics where the cost of introducing a new
technology declines over time and the technology can migrate between countries. Higher growth rates in the follower country combined with a maturing of the advanced country’s market then permits the follower’s costs and quality to catch up through greater productivity increases and higher growth (Rapp 1973). However, user base economics constantly favor the advanced country with a large user base, especially if this is extended globally. This is because the cost of development of a new software product or upgrade is essentially a fixed cost and is very large; while variable costs are low. Thus, the larger the user base, which is a function of the number and size of one’s customers, the smaller is the fixed cost per unit. Potential productivity differentials between countries are overwhelmed by the effects of user base differences (fixed costs divided by a large number of users). Users in this sense are the actual end users of a software program. In the case of an operating system, it is the user of the machine; in the case of programming languages, it is the programmers; in the case of application programs, it is the person developing the program or using it to accomplish a specific task. Thus, within a particular large customer, there will be several users depending on the type of software. However, once the users have a program that accomplishes their needs, and they have developed the skill set to use it or improve it, there is little incentive for them or their employer (customer) to switch to a competing system. For this reason, substantial penetration of the existing customer base has usually occurred only when new hardware and software were purchased to address a new problem the existing system could not accommodate. Even so, over the last thirty years, large Japanese corporations have experienced many such situations so that while their mainframe system may still be primarily Fujitsu, Hitachi or IBM they now have other systems such as Hewlett-Packard and DEC as well, purchased for
special reasons, as well as collections of different PCs. They now want to link these together, offering opportunities to network systems producers. Yet, the network solutions must also maintain the customers’ proprietary systems.

This continuing fragmentation and the preservation of customized solutions has forced MITI to search for new, more successful policy approaches to encourage the packaged software industry’s development. What is not clear is whether a new policy model will address the realities of the industry’s evolution, its economic structure, or the reasons the established fast-follower development 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 this set (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 customer’s mainframe purchase decision through the foreign exchange law and import licenses. Technology import licenses and administrative guidance were used to contain 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 US government policy involvement. MITI can no longer pressure companies to buy Japanese software or switch systems contrary to their interests. The US government has also
taken a more proactive stance in objecting to particular actions such as mandating the use of TRON in Japanese schools (Coultas 1994), changing intellectual property laws protecting software (Rapp 1986), or recent initiatives concerning decompilation or the introduction of an ISO 9000 standard for software sold in Japan.

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 such as Nippon Steel and Toyota can recruit the software engineers they need to maintain and develop their proprietary systems, often assigning them to their captive software subsidiaries. The growth of these companies’ share in a relatively flat software market is indicative of this trend (JISA 1993). Therefore, few are left over to support a large independent packaged software industry. In software, Porter’s (1990) positive agglomeration effects do not seem to exist.

The major integrated suppliers that produce 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 packaged software, just as they did with government help in the 1960s and 70s.

In fact, the integrated suppliers are actively supporting the localization and adaptation of foreign application software to their systems (AEA 1992; Rapp 1993A), increasing US firms’ user base, lowering their per unit costs, and improving their global market share. Nor are integrated system suppliers reallocating scarce technical development personnel from hardware to software. Their strategic emphasis remains hardware.

Why are Japanese integrated system producers and large customers 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? The application of evolutionary theory indicates that each market segment has evolved differently, and the segments are not growing equally. In respect to the table below, mainframe software developed first, then software for stand-alone minis and more recently for network servers (usually mini computers that manage the network using a communications management program), workstations and PCs. As a large but mature market, mainframe sales and software are growing much less rapidly than the market for network servers, work stations, and PCs and their related software. Further, mainframes are used more for distributed processing, data base management, security, large computational requirements and overall Electronic Data Processing (EDP) than for specific task-oriented activities which appear to be customers’ current interest as they try to put more computing power throughout their organizations and into the hands of the ultimate in-house user.
Data on the size and growth of the Japanese software market by equipment segment, and whether packaged or customized, appear in Tables 1 and 2. MITI does the most comprehensive survey of the software industry (MITI 1993) The Japanese Information Service Industry Association (JISA 1993) uses MITI survey data and also surveys a sample of predominantly larger software suppliers to obtain answers to more specific questions. Dataquest, a Dun & Bradstreet division, is an independent estimator of software/semiconductor and other information industry markets globally and by country, including Japan.

The MITI estimate is Y4.3 trillion per year for 1991-92 in software industry turnover given in its Specific Service Industry Fact Finding Survey: Information Services Industry Version (MITI 1993) of which 85% is customized. This substantially exceeds the sales figures estimated by Dataquest (Boyd 1995) for pure software in 1993 and 1994 of Y2.4 trillion; however Dataquest estimates the same percentage of customization. Even though the total size of the software market appears to be different from the two sources, the market share of customization does not. The difference in market size are due both to coverage and to differences in the definition of software used.

Table 2 indicates large customers have a relatively greater commitment to customization than the market as a whole, around 95% versus 85%, though these large scale providers and their customers still account for 5-10% of the packaged market including PC software. The customized market’s dependence on larger firms is not surprising given that about 50% of the packaged market is PC related (AEA July 1992) and that smaller firms and individual users do not buy mainframes but are a big market for PCs. What is unusual is the continued growth and dominance of customization in the overall market and within the large customer segment even for PC software. It should be noted that these figures do not include integrated or processing services, which can include software delivery. These large systems providers have about 20% of that market. In any case, MITI’s figures seem to represent the upper bound on the Japanese software market’s size and the importance of customized software within it, while the JISA (1993) data indicate the relative growth of large customer demand and large systems providers, including captives, within an overall stagnant market.)

Table 1 Estimated Japanese Software Market Size by Segment 1992
(Yen billions; percent)

|------------------|----------------------------|

46
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Customized Sales</td>
<td>Y562</td>
<td>12.2%</td>
<td>7.1%</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>46.0</td>
<td>18.9</td>
<td>10.3</td>
</tr>
</tbody>
</table>

(Notes: This table represents a synthesis of MITI (1993) and JISA (1993) estimates. The 85/15 custom/package split is also supported by the JISA (1993) EDP expenditure survey.)

Table 2 Results of JISA Survey (1993) of Large Japanese Customer Software Market:
Relative Size by Market Segment and Growth Rates
Types of Software

The software market is broken down into software associated and used with particular hardware: super computers, mainframes, network servers (including minis and office computers), workstations, PCs, and game machines. It is also separated within each of these categories between customized and packaged: the former is developed for a particular use and customer, the latter is generally applicable to a wide range of users and customers such as word processing, spreadsheet and game programs. There is another set of strategically important distinctions as between the instruction set, operating systems, middleware, tools and application programs.

Instruction sets are the instructions hardwired into the machine. It is, for example, what makes a machine IBM compatible. Operating systems are the software used to run various computers, i.e. instructions such as MS-DOS (MicroSoft Digital Operating System) or MAC that interfaces with the instruction set so that the machine can understand and act upon instructions given to it by the middleware and application programs. As identified by the AEA (July 1992), the most important of the various incompatible operating systems used in Japan are: mainframes (IBM-MVS, VM & VSE; Fujitsu-MSP & FSP; Hitachi-VOS 1 & VOS 3; NEC-ACOS 4 & ACOS 6), minis (IBM-
OS/400; DEC-VMS; Hewlett-Packard-HP UX), workstations (UNIX System V), and PCs (IBM-
MS DOS & IBM DOS 5.0/V, Microsoft-MS DOS 5.0/V and NEC—until recently an incompatible
version of MS-DOS).

Therefore, firms have developed middleware to permit different application programs to
work on these disparate systems in a multivendor environment where large customers want their
different hardware/software configurations to communicate. Middleware thus stands between the
operating systems and the network software or various application programs. It contains the
various standard protocols and communications programs that allow programs written for other
operating systems or on other machines to communicate and interact. Networking programs are
thus a type of middleware. The middleware also provides the Application Program Interface
(API) that lets application programmers write programs to that interface which will then run on a
particular machine with a particular operating system. Application programs are those written to
perform specific tasks and in Japan applications are usually written and designed by
programmers and systems engineers for the large customers as part of their proprietary software
systems. There about 280,000 such individuals working in the Japanese information service
industry, representing a little more than half the total people employed (MITI 1993).
Programmers use programming languages such as PASCAL, COBOL, FORTRAN, BASIC, or
C++ to write application programs. Each popular language in turn writes to the API for a
particular operating system. So there is a C++ for MS-DOS as well as for MAC. In Japan, most
applications are written for mainframes in COBOL. After the program is written, a compiler
program translates it in the zeros and ones that can be understood by the machine. It is thus
machine or microprocessor (MPU) dependent. Decompilation is the technique for reversing this process in order to understand how a particular program has been written, and decompilation is a current area of policy dispute in Japan, the United States and elsewhere regarding intellectual property.

Since in Japan the operating systems for mainframes have their unique historical characteristics and the common operating systems for network servers, workstations and PCs are different as well, writing middleware that allows one’s operating system to communicate through a network to a large number of different operating systems has become a key strategic variable for US firms pursuing a hub and spoke strategy to get customers to buy their software. Hewlett-Parkard, for example, has its own unique operating system that runs its network servers and workstations. But it needs to be able to integrate its system into the proprietary systems of its major customers who have other computers with multiple and distinct operating systems from H-P’s, as well as highly customized application programs that run on those other systems. To enable their system to act as a communications link between these various computers and software programs and for their customers to run their application programs through H-P’s network servers or at their workstations, they must provide the appropriate middleware. Thus middleware has become the plug of a plug compatible computer system. While H-P provides both operating systems and middleware, it supplies only up to the API; it does not supply application programs since generally clients develop these themselves.

Several foreign software developers have taken advantage of this customer demand for system integration and interoperability to further improve their cost competitiveness by
exploiting a hub and spoke strategy that maximizes their 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. But there is little competition to shift core clients from their mainframe systems or operating platforms; so those market shares remain stable.

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 Japanese integrated systems producers’ differentiated operating systems represent the spokes along which the foreign producers market to the systems producers’ captive mainframe client base. The integrated systems producer’s strategy is to maintain their Japanese customer base by supplying hardware and software for all segments (mainframe, servers etc.) to existing customers even to the extent of marketing foreign software packages. In sum, each such producer and their operating system represent a spoke, with the mainframe system at the intersection of the spoke with the outside rim. The independent packaged software developer’s strategy is to try to sell into as many of these client bases (spokes) as possible through differentiated adaptation, generally through writing specialized middleware with the cooperation of the integrated systems developer. At the last stage, the software is customized for the individual users. To maintain confidentiality, large clients do most of this customization themselves through their captive software development subsidiaries.
Writing this type of middleware, and working with clients to install it so that it really links the entire diverse system of software and various hardware together, has thus become a critical strategy factor in selling hardware into a multivendor multi-operating system environment. It is also very expensive. This may be why Fujitsu has recently announced its decision to sell some of its middleware solutions globally. That is, as part of its strategy is to sell its hardware into a multi-vendor global market, it must develop the appropriate middleware. However, to reduce the unit cost of this middleware, thus making Fujitsu’s total system less costly to their hardware clients, it is trying to expand the user base for its middleware by selling this middleware independently in the global market. While middleware is likely to be highly customized to meet individual customers’ specific systems requirements, the operating systems and language programs are usually packaged and preinstalled in the computer. According to Dataquest, in 1994 system software represented about 4% of the market while tools and relational data base management systems represented about 6% and 4% respectively (Boyd 1995). However, they account for about 60% of the packaged software sold in Japan (AEA July 1992). This means that application packaged software is only about 5-6% of the total Japanese software market and 6-7% of the Japanese application software market segment. Customization’s heavy application orientation is fully consistent with the interview and questionnaire results and is another indication of large customers’ commitment to using software as part of controlling their business processes and improving productivity.

**ECONOMICS OF SOFTWARE**
The basic economic facts of software development and production are that development costs are high and fixed and reproduction and packaging costs are very low and incremental variable costs. There are two basic approaches to analyzing the cost of software development. One is a production model based on improving the efficiency and quality of producing a line of code. The second is dependent on the number of users and those users’ commitment to a particular system, program or software solution. The first might be described as a production based strategy whereas the latter as described above is a user based strategy. The following discussion first considers Japanese producers’ approach to software development as highly customized; the second part evaluates the logic of large buyers’ commitment to customization, especially for application programs, when combined with the advantages foreign firms with larger user bases have in serving the packaged software market.

**Producer Economics**

As described in Cusumano’s book on the Japanese Software Factory (1991), several software firms, both US and Japanese, over the years 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 in Japan because it works as a way to reduce costs in developing customized software or adapting and localizing foreign software.

These procedures have facilitated continuous improvement in existing programs in terms of code written per hour and in 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 useable program, an important benefit since these technical personnel are in scarce supply. Indeed, the aging of the Japanese population means that there will be fewer new labor force entrants to become technically trained in these activities in the future. By breaking the task down into discrete modules, regular employees with less training or even contract personnel can do the work. Further, industry commonalty 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 (Imai 1986). This has been demonstrated, however, only for larger firms (Cusumano 1991). It is not clear if such results extend to smaller firms. Certainly, it has not extended to the competitive development of new software programs or the types of middleware solutions described above.

In the early 1990s, there appeared several similarities between software production techniques applied by large integrated electronics producers such as Hitachi and Fujitsu and the techniques they used to produce integrated circuits (ICs). Managers logically acted on the basis of past success. The government also supported the industry using similar cooperative development programs but with a policy emphasis on common standards, concentration of resources, and high market shares, all proven techniques to reduce cost and improve quality among select firms in many manufacturing sectors which could 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,
Japanese managers in the large integrated systems producers who are also among Japan’s leading IC producers have now shifted to a hub and spoke structure. They are adapting foreign software to their clients’ proprietary software systems instead (Toka 1992).

Thus, because of this shift in perspective and because it would also necessitate substantial program revisions, TRON has not been accepted by the Japanese software industry, large customer firms, 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 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 Japanese firms to adopt it since it would be high cost and of limited use. Rather, using TRON as an operating system 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 could mean loss of their proprietary production systems. Both choices are unattractive to Japanese corporate customers compared to the existing situation. Therefore, they have not supported it.

In looking at successful Japanese competition in industries such as semiconductors, autos, and consumer electronics, one constantly repeated theme is that managing growth is the key to increased market share and firm survival. This approach is most successful in assembly industries where a standard product is produced on a repetitive basis involving hundreds or even thousands of parts and using complex production tasks, the opportunities for continuous integration are great. However, benefits in applying these rules and procedures to software

55
development and competitiveness appear questionable, due to the differences between software and semiconductors in industry structure, fixed and variable costs, and competitive dynamics. The role of timing is also important though. US firms retained their lead in semiconductors as long as US 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 DRAM market 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 product, but if growth slows, it becomes more difficult, as Lotus discovered. So firms cut back, giving Japanese companies and other competitors 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 for follower companies to increase market share and costly to unseat established leaders. Because the mainframe market in Japan is mature, market fragmentation is likely to persist for mainframes. Indeed, most customers and industry participants in this project did not even discuss competition for market share in mainframes as an issue, though the recent IBM agreement to sell mainframes and PCs to Hitachi on a OEM basis may have somewhat upset this balance. However, that is unlikely to affect the mainframe software market since the differential operating systems will persist. Given the need for system integration, the disparate situation in mainframes forces customization in the new high growth segments such as 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 software producers could easily leverage their domestic position into an aggressive expansion strategy during a downturn, while their localization skills are not necessarily expandable to develop products salable in global markets; at the same time, US firms appear to be capturing the global growth. Indeed, systems development for international sales has proved difficult for Japanese firms—even in areas such as cellular telephone systems where they have competitive hardware—because Japanese operating systems have not easily adapted to overseas billing and record keeping requirements.

**Customer Economics**

Emergence of a standard or dominant design might favor Japanese producers, but it must be demanded by Japanese customers. However, there seems to be no strong trend in this direction for software. Rather, economics and administrative heritage argue customization will persist. In consumer electronics, the effects of multimedia and giant global networks may force standards for these products, and Japanese hardware firms should be well positioned to benefit from this development no matter how it evolves. Nonetheless, they will find plenty of competition from the Korean firms and a resurgent American industry. The operating and application programs related to multimedia are likely to remain American, though. For example, Microsoft has written the operating system for the new Sega Saturn CD-ROM based game machine (Choy 1994). However, Japanese firms will continue to do well in games software per se..

Such administrative heritage in large client firms and their industries 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 focus on developing software for the steel industry (Baba et al 1993) It is in the manufacturing 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 is based on highly integrated computer systems that begin with the order from the customer, then schedule the production run, notify the appropriate suppliers, and on through the production cycle. In this respect Japan has a very successful and competitive international software industry, but its competitiveness emerges in the customer’s product -- a car -- rather than as a shrink-wrapped software package. This specialized software is not costly relative to total production costs; for example, EDP costs are less than $100 per car. Any cost savings in software, which in any case represents only a portion of total EDP expenditures, would be relatively minor compared to the possible disadvantages of not using their proprietary systems, and the possible downtime from incorporating a totally new production system rather than making incremental improvements to the existing one.

Seen in this light, leading Japanese firms' emphasis on customization makes sense since software is an integral part of a successful integrated production system. Indeed, it may be an integral part of a successful global business strategy. 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 unit of output 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). Indeed, from this vantage point it may be that though the Japanese are not leaders in developing packaged software, they are leaders in improving and using software as a strategic tool to advance their overall business objectives. After all, no US auto company has developed a computerized ordering system that can deliver a car produced to the customer’s specifications in two to three weeks while eliminating the costs associated with expensive excess inventory, end-of-year sales and dealer floor-planning.

On the other hand, buying existing software that has been localized, customized and debugged in many cases offers a reasonable alternative to full customization while retaining most of its benefits. Based on the interviews and questionnaires, while customization can triple the cost of installing a package as a semi-customized solution, full customization can run ten to fifteen times the cost of a package. Companies may not view the semi-customized solution as so expensive relative to developing a comprehensive new system from scratch, that is their real alternative, not the purchase and use of a packaged system with little modification. This is because the package will not work as part of their proprietary information systems and so is not a productive expenditure. This trend of course creates a big advantage for foreign suppliers of packaged software that can be inexpensively localized, as is already the case for many foreign software developers. This is especially for ones that have experience adopting it to different operating systems through the effective implementation of a hub and spoke strategy.

One implication of this scenario is that where the need for customization is the strongest, namely in mainframes and then in network systems, the customized market should be much
larger than the packaged market. This is due to the fact that many packages will be semicustomized at a cost of more than twice the package program itself. Conversely, in a market such as PC software, where there are more stand alone packages such as word processing programs that do not need to be customized, the packaged market should be relatively larger compared to the customized market. The data in the tables above on the software sales of large systems integrators to their large clients and the software market as a whole indicates this pattern exists for the Japanese software market, thus supporting the hypothesis.

The extensive integration of software into the large customers’ business system is a major reason why Japanese large customer firms have maintained a strong mainframe mentality, allowing the market for mainframes and their software to persist more strongly than in other developed markets. In 1991, about 50% of computer shipments in Japan were still mainframe (JIPDEC 1993). Because they are risk adverse and because of the high degree of integration, large firms want to stay close to their existing technologies, products, and ways of doing business; 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 are the way to achieve competitive advantage, beginning with what one knows will work (Imai 1986). In this technical and cultural 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 who are former and perhaps future managers of business units 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. Indeed, the most automated production system in Japan, Nissan's Zama plant, has been shut down in the current recession; GM's highly automated facility at Lordsville was not a success either.

In this case, the arguments of economists such as Aoki (1988, 1987) that Japanese competitiveness flows from its organizational differences seem applicable when extended on an intra-industry basis, i.e. within steel and autos, but not on an inter-industry one such as the migration of technological competitiveness to packaged software. The organization of Japanese industry has actually frustrated the development of a globally competitive Japanese packaged software industry while it has promoted successful customization of software to the needs of its leading software customers. They in turn have maintained and extended their global competitiveness by constantly improving productivity. They see part of their ability to do this as related to their customized software strategy. Still, even among major customers there is much room for improvement beyond the shop floor. Compensation and promotion based on seniority have retarded adoption of computers in the office, slowing the productivity growth of white-collar workers. Until recently, management has felt little competitive pressure to introduce newer software technologies to make office workers more productive. In addition, the downsizing and reengineering that often justifies the introduction of networks and workstation/PC systems is not always an immediate cost benefit in large Japanese firms. This is because the permanent employment system lengthens the period during which such rationalization will occur.
This differs greatly from the situation in the US where job mobility is high, where computer literacy can now be an important executive requirement, and hence where rationalization can be implemented much more swiftly. For instance, 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 US. This in turn has supported the evolution and growth of packaged software. In contrast, job stability promotes and facilitates customization in Japan, since workers can both spend the time to fully learn the proprietary system and feel little risk in doing so. Interviewees in both Japan and the US 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. But such large scale organizational change appears to occur more readily in the US.

Maintaining US competitiveness in packaged software thus seems to depend on the continuation and promotion of certain management practices and institutional arrangements which enhance flexibility and change. While certain practices and arrangements have worked to the advantage of Japanese firms in many industries, in software they are not so helpful, though apparently benefiting American firms which operate in a very different managerial environment. This counters Aoki’s conclusion that management success is not industry specific. Ironically, successful cooperation within firms and between Japanese and foreign firms has been important in achieving the foreign packaged software producers’ overall business objectives in Japan, where it is currently estimated they hold 75% of the market (JEI 1994). 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, then large customers’ internal decisions will insure that the Japanese software industry’s fragmentation will persist. At the same time, the willingness of the large Japanese integrated systems producers to work with the foreign package software suppliers to upgrade their clients’ proprietary systems means that semi-customization will create a constant expansion of the foreign suppliers’ Japanese and global user base, improving further their ability to compete in Japan. In this way, observing differences in current management procedures, an understanding the origins of those practices and motivations behind them, has helped the research team comprehend 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 US to achieve a transfer of skills, as happened earlier in various manufacturing industries. However, given that US firms are concerned with technology loss and do not seem to need to build new linkages with Japanese systems engineers to open the Japanese market, it is unlikely this will occur on a large scale.

The Economics of Packaged Software, Including Games

It is apparent to most participants in the packaged software industry, as distinct from the customized market, that for them software industry economics differs from manufacturing in ways that effect the motivation and behavior of producers and customers. As explained above, the bulk of a software program’s cost is in the development stage rather than the physical
manufacture and hence are fixed costs. Actual production and distribution is cheap (Steinmueller 1993) 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 so as to reduce fixed costs per unit of sales.

For packaged software, it is the size of the user base or cumulative units installed, not the cost of production, that 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 when the ability to easily share software based information across users is important, 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, graphics program or Internet searches), leading to an ever larger user base and lower prices, with generally 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 software programs which manage and handle the systems of
their largest competitors in order to capture the externalities of increased user interchange (e.g. Microsoft and Novell). 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 US packaged software companies have developed source codes that are compatible with foreign language translation to facilitate this process.

Given this structure copyright, while normally considered a weak form of intellectual property protection, is actually a good way to protect packaged software on a long term basis. This is because protection in global markets is automatic and for the life of the corporation plus fifty years. There are also no expensive filing requirements as in the case of patents. Yet, because it is difficult to change the expression of the programming idea without changing the external benefits to the user, it means the infringer does not have ready access to the user base that drives firm economics. Nevertheless, because foreign companies are the main beneficiaries of copyright protection and because Japanese developers primarily produce customized software that have other innate forms of protection, there are not great pressures in Japan for intellectual property (IP) enforcement except perhaps from game producers. Software piracy in Japan remains a problem, hurting developers’ potential cash flow. However, for the foreign developer, piracy also has the more sanguine effect of undermining potential or actual local Japanese competitors.
Further, users gained through piracy may become customers through better IP enforcement in the future.

The unique features of the industry have interacted with different historical legacies and competitive environments in Japan and the US to place the Japanese and American software industries on different evolutionary paths that seem likely to persist. Because large customers and their multiple users have developed and maintained highly customized systems, the favorable economics of the software industry of a large user base with positive externalities that exist in the US do not exist in Japan, except for foreign software developers, game producers, and one or two packaged software producers. This limits the public-good character of software in Japan, unlike the US. That is, in the US, software use by one person does not diminish its use by someone else (H. See 1992 and S. La Croix 1992) and indeed its utility and value may be enhanced by increased usage, in Japan its value appears diminished because a firm’s proprietary customized software system is considered part of its competitive advantage. This benefit would be lost if someone else were able to share it. So while in the US and elsewhere, 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 this is only a benefit within the firm and to some degree within the vertical group.

Moreover, since there is little employee migration between large firms, Japanese companies feel little need for systems compatibility across firms; indeed, shared software usage would reveal sensitive competitive information. Since software in Japan does not reflect aspects of public goods’ positive externalities, there are then fewer economic pressures towards merger,
integration or consolidation. Rapid per unit cost reduction is not the norm, and indeed not possible except in games; it is limited to the development of a few related programs applicable within an industry or company group. As explained above, this creates pressures to reduce cost via reusable program segments (Cusumano 1991) and by the establishment of captive software subsidiaries and affiliates. JISA’s 1992 survey of large software customers reveals that among firms with more than Y500 billion in revenues 54% have or plan to have a software subsidiary or group related affiliate (JISA 1993) compared to 28% for firms over Y100 billion and 23% for those over Y50 billion.

Offsetting the resulting high unit software costs are the competitive advantages conveyed by a firm’s proprietary innovative processes, the basis on which Japanese firms generally compete domestically and with foreign competitors which are more likely to emphasize product innovation than process innovation (Womack, Jones, and Roos 1990 and Rapp 1992). Under these circumstances, there is little change in market share among the leading integrated computer systems producers for mainframe customers (Rapp 1993A); this helps them maintain both customers and sales. The leading global competitors in emerging high growth markets, such as network servers and work stations tend to be the global leaders. For example, Sun and H-P have over 50% of the market in Japan for workstations (AEA July 1992), while sale of competing Japanese systems like NEC’s are sold mostly to their existing Japanese mainframe customers.

In games, by contrast, market share can be increased 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; in either case new machines do not play the old
cartridges. Thus, unlike large Japanese software customers, there is rapid obsolescence, and compatibility and substitution with an existing system are not competitive barriers for game producers.

Furthermore, in the game segment the software is the key component of the end product - not a ton of steel, an automobile or an insurance policy. As such, there is no organizational risk to the customer, typically an individual or family, in switching to a new system while differences in the format and play between systems from a users’ standpoint appear 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 US producers such as 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 dynamics and outcomes become quite different.

This raises the question whether Japanese software suppliers to the steel and automobile industries should be considered part of the manufacturing industry or as software developers. We take the position that they are really part of the user industry, not a distinctive software industry. These suppliers are usually owned by the customers and staffed by their 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 their user base. Finally, they are dependent on the economic health of their customer’s industry for their own survival. The results of the industry survey conducted for this project indicate that most software developers working for large companies
are so driven by those companies’ organization and demands that there really seems to be no autonomous Japanese software industry per se, except for games.

Because Japanese firms are not the global leaders in most software other than games, US firms control the migration of competitive advantage in programming and usage to countries such as India and Russia, where they use low cost, highly trained talent. Japan is now a high cost country for skilled technical labor such as computer engineering and systems programming. To become competitive it needs to make software development more capital or machine intensive, though so far this substitution approach has proved elusive. Educational and firm infrastructure also exert unusually strong influences on software development. But in this case the market has not promoted the strong development of the “advanced factors” of production such as software engineers and programming specialists needed to develop the industry rapidly. Therefore, the development and expansion in Japan of the type of agglomerative structure which Porter notes is necessary to achieve and maintain competitiveness (1990) in an industry sector is not present in software unlike steel, autos and electronics for example.

**HARDWARE, TECHNOLOGY AND MARKET SEGMENTS**

A related factor favoring US developers and continued fragmentation of Japan’s software industry is the strong relationship in the market between particular operating software and certain microprocessors (MPUs) or logic devices that drive computers, since US producers dominate the global market for microprocessors. Specific operating systems and related software have frequently been developed to work with certain MPUs, and software programmers are constantly developing software to take advantage of the next generation of more powerful, faster MPUs.
There are strong market forces linking MPU producers, programmers, and users; they 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 be able to run the old software (operating systems, middleware and application programs) as well as the new. Similarly, software developers want to be able to use their existing or a closely related programming languages to upgrade operating systems, middleware and application programs as this increases productivity, improves quality and reduces development time.

Software developers will concentrate on developing new programs and upgrades for the most popular MPUs, which due to their high development costs are subject to the same user base (fixed cost) economics as packaged software. This has led to an interactive oligopolistic cycle, combined with the growth in personal computers, networking, and parallel processing, to make MPUs and their related software the defining factors shaping competition in specific software markets. The examples are for PCs Intel and Motorola; for workstations Sun and Hewlett-Packard; and for mainframes IBM, DEC, Intel, and Motorola. This trend probably will not change in the foreseeable future; indeed, it seems to be getting stronger following the introduction of the Power PC and Pentium processors. That is, while MAC and IBM/PC based chips and software dominate PCs, UNIX based systems are preeminent in workstations, currently the fastest growing Japanese market segment. IBM compatible systems still lead in mainframes. There are no comparable Japanese global leaders for MPUs and their related instruction sets in any of these market segments.
With these multiple linkages, there is no single MPU/software interaction upon which Japanese integrated systems producers or software developers can focus. Even among the various generic operating systems, each firm has had its own unique mode. While this has the advantage for the Japanese hardware and software suppliers of restricting customer migration, it discourages the development in Japan of the positive externalities that would benefit the whole Japanese software industry if there were a single a dominant MPU/software nexus on which they could concentrate.

Also contributing to this situation is the fact that US MPU producers learned about the importance of Japanese market share from the US experience in steel, autos and DRAMs. They thus entered the Japanese market for MPUs earlier. Indeed, Japanese firms’ emphasis on DRAM production combined with the US Semiconductor Accord in 1984 served to increase the demand for US MPUs since Japanese firms committed to buying a certain share of US made chips. Because Japanese companies did not yet produce MPUs in quantity, it was easiest to purchase the US made MPUs. This pulled US suppliers into the market. In turn, Japanese firms underestimated the eventual strategic importance of designing and producing MPUs. They mistakenly assumed that, as had occurred other industries, their manufacturing technology would be better and they could ultimately move into logic chips through licensing once they had achieved dominance as the low cost chip producers via DRAMs. US producers, however, had learned through earlier experience and applied a new strategy. They established their own production facilities in Japan, and they did not license their technology to Japanese firms. In
addition, they vigorously enforced their intellectual property rights as exemplified by Intel’s case against NEC.

Accordingly, US producers were well positioned when shifts in computer technology and market growth towards parallel processing and PCs combined with multiple generations of faster microprocessors (e.g. 16k to 32k to 64K bytes per second) to change the standard dynamic. They also benefited from a stronger yen, the result of Japan’s previous competitive successes. The development of the silicon factory and customized chip producers also moved production economics against Japanese expectations. Profits turned out to be in logic and design rather than from efficient standardized production; and the shift from producing memory chips to designing and producing MPUs turned out not to be so easy. The very regularity of the DRAM meant design capability was not emphasized by Japanese firms. The role of history and administrative heritage in this competitive process is thus clear with evolutionary analysis offering the best explanation of future trends.

The fifth generation computer project attempted to replicate the VLSI project with respect to software. But of the various planned projects, the only large development that succeeded was the designing of a computer capable of handling inference programs, i.e. a hardware not a software development. Further, most productivity improvements in software production over the last few years, such as object-oriented software, have been primarily software based developments pioneered and extended primarily by US developers. Machines capable of writing software have yet to be produced.
In addition, as noted above, an important part of software development is to maintain compatibility with existing programs and operating systems, since users do not want to relearn or rewrite particular programs every time they purchase a new machine. Under these circumstances, the most important competitive factor for a software producer is its installed customer base and existing program design or library, combined with its ability to write new programs or upgrades for existing programs for new or improved hardware so that changes are relatively seamless and easy to managed from the users’ viewpoint. Apparently, maintaining this interface while offering many new features was one reason for the long launch time of Windows 95. In this competitive environment, competitive advantage is therefore determined by these other factors and not, as initially hypothesized by Cusumano (1991), by the ability to produce more error free line code in a given time period.

The cost of line code per user is a more important variable 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 dramatically. When two programs with the same function and roughly the same number of lines of codes are in competition, the cost competitiveness will be directly tied to their respective installed user 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, or how new niche companies can quickly establish dominance in a particular area. 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 large recent US software acquisitions reflect and confirm these cost factors.

If Japanese firms were interested in responding to these competitive conditions, they have several traditional options. They can wait for US growth to slow and for US firms to establish a standard dominant design they can easily emulate. But, there are difficulties with this strategy. Software control through copyright is for 55 years or more, so Japanese firms cannot wait. It is not clear that any dominant design effective across all market segments will be established, given the presence of different leading MPUs and operating systems in different market segments. Moreover, Japan itself is a relatively large and mature software market; by the time US growth slows, it will be slowing in Japan too. Another possible strategy is more cooperation.

In fact, these cost and intellectual property realities have combined with continued customization and fragmentation to produce more cooperative strategic approaches by both Japanese and US firms. There have been significant increases in strategic joint venturing such as the tie-ups between Toshiba and Motorola in logic chip production, Sun and Fujitsu in RISC production and workstations, and Apple and Fujitsu in multimedia. While joint ventures limit a Japanese firm’s ability to capture new markets, it helps the firm retain its domestic customer base. One more way has been the Japanese government’s 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 their original zeroes and ones and then rebuild them into usable programs has evoked a strong US government policy reaction.
In practice, though, decompilation may do little to change the user base advantage of the foreign developers. It represents just another form of localization that siphons off programming resources from development to copying and localizing the copied product. This type of re-engineering results in "legalized dumping" of localized US software onto the Japanese market. De facto, it continues US dominance of software and hinders the evolution of an independent Japanese packaged 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 whereby hardware producers have organized themselves to assist foreign independent software developers in localizing their software so that a shortage of attractive software does not threaten their hardware client base. This only gives further impetus to the trend that pushes Japan's programming resources towards localization and adaptation in packaged software rather than towards independent development.

With the exception of Sega and Nintendo in games, there has emerged no dominant Japanese firm with a global software presence. 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 and foreign 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 packaged software design despite MITI's efforts to do this for UNIX via the Sigma project.
INDUSTRY STRUCTURE

Mainframes

The Japanese mainframe software market is large: estimated at Yen 2409 billion in 1992, of which Yen 2233 billion is customized, with growth for 1991-94 among large customers expected at 5.0% per year despite the leveling off or decline in the number of mainframes sold. Thus, customized mainframe software sales will be relatively stable (see Table 3 below). The high growth opportunities in selling to large customers should be in downsizing, i.e. network servers and workstations, and related middleware and application software, both custom and packaged. At the same time, as Table 1 shows, the customized mainframe software market is larger than all other software markets combined and is likely to remain so for a while, absorbing considerable financial and personnel resources. In addition, some customers are using mainframes as large network servers in addition to their role as the “host” computer.

Table 3 Growth Rates for Japanese Software by Market Segment 1990-94 - JISA

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Mainframe software</td>
<td>5.5%</td>
<td>5.0</td>
<td>-8.8</td>
<td>7.2</td>
</tr>
<tr>
<td>Mini Computer software</td>
<td>22.1</td>
<td>7.3</td>
<td>36.3</td>
<td>11.8</td>
</tr>
<tr>
<td>Work Station software</td>
<td>33.9</td>
<td>13.3</td>
<td>18.2</td>
<td>34.7</td>
</tr>
<tr>
<td>PC software</td>
<td>18.9</td>
<td>10.3</td>
<td>3.7</td>
<td>12.0</td>
</tr>
</tbody>
</table>

Source: JISA (1993)

This view that the importance of customized mainframe software will continue is confirmed by the interviews with large customers. They generally agreed it would be five years
before they had fully converted the various tasks that they presently envision changing to
network servers, work stations and PCs. Even then, they forecast that 30% to 50% of their total
EDP costs will remain mainframe-related and the associated software would continue to be
customized.

Office Computers, Minis and Network Servers, Workstations and PCs

A 1992 MITI survey (reported in JISA 1993) 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 customers want the new software to be
integrated into the company’s overall system. Customers indicate that, except for word-
processing and spread sheet programs, almost all the packaged software they bought, even for
PCs and workstations much less for network servers, stand alone minis or mainframes, had to be
adapted to their systems. This is true despite customization expenses that can run two to three
times the cost of similar localized packaged software. In other words, customizing costs comprise
around 70% of the final cost of installing the package. Nevertheless, this is cost effective
compared to full customization, which can run more than ten times the cost of a localized
packaged program.

From the available data and this analysis, it is clear that, although highly visible, shrink
wrapped packaged PC software is a very small part of the Japanese software market, about 7-8%,
and depending on the customer and distribution segment can be as low as 3%. Nor does this
seem likely change for the foreseeable future despite the substantial increases in sales of PCs due
to the attitudes of the large software customers. Customized mainframe software will continue to
represent the bulk of the market and the main user of software resources. This conclusion is at odds with the convergence theorem put forth by many technical analysts and supported by Martin Fransman in his to be published book on the Japanese computer industry (1995) based on the growth in software demand during the bubble period. It also runs counter to the AEA (July 1992) study’s projection for a more than doubling in package software sales between 1990 and 1995 or about a 20% compounded annual growth rate. Yet, as the Dataquest (Boyd 1995) and MITI figures show, software sales including total packaged software sales have grown only modestly during this period, and there has been no shift in market share relative to customized software. Therefore, even though Dataquest is predicting that this shift will now finally take place between 1995 and 1998, despite a continuing slow growth market, based on the interviews, questionnaires and industry economics we remain skeptical concerning these projections. Our forecast is therefore that customization and semi-customization will persist at relatively higher levels due to the large customers’ economic commitment to their proprietary systems and the establishment of their captive software subsidiaries.

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. They are now being perpetuated because of the high cost and organizational difficulty to customers and their corporate users of significantly changing their existing systems, combined with their suppliers’ strategic interest in maintaining their client base. This situation persists despite even though continuously customizing software is uniformly recognized by large customers 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, though most firms do not expect them to be less expensive solutions.

In this environment, the strategic issues in this industry have shifted from a producer to a customer driven model. However, MITI’s historical development role has generally been to support producers, not to change customer 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 here 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), customers are only concerned with their own businesses and industry competitors. They are interested in the contribution information systems can make to their competitiveness. For them software is an input, not an output.

The technical contribution of that 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 research paper is good or bad whether it is
produced by pencil and paper or a sophisticated word processing program. Japanese customers see little benefit to supporting a Japanese produced rather than a foreign 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 large firms are interested in controlling.

Viewed from this perspective, important aspects of customization come into focus. As described above, leading Japanese industrial 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).

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 customer interests as well. It has done this by formalizing the uniqueness of a firm’s competitive advantage and making it impossible for competitors to gain access to their total information system. Thus security and control are important aspects of customization as well as the sunk costs in training,
hardware and line code. Having selected this decision rule and seen it work, the firm logically continues to pursue it. 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, the information system cost per car is only around Y10,000, or 1-1.5% of the ex factory, very small relative to the importance of maintaining Toyota’s production and organizational technology advantage. The cost savings of using packaged software are thus not seen as large relative to the productivity gains of customization. There is therefore a symbiotic relationship between the “captive” customer 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, customers still must keep their systems technologically current. Often this means using more advanced or specialized foreign software. However, customers 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 is essentially an incremental approach to software innovation which works well unless there is a profound technological transformation that completely alters the cost structure. This approach once again reflects and demonstrates Japanese users’ process orientation and administrative heritage (AEA 1992). In positioning foreign software for such
adoption into a large firm’s customized proprietary system, there are three steps necessary to sell the software: localization to Japanese language and format, conversion to the customer’s mainframe platform, and customization to its specific process and requirements. Based on the questionnaires and interviews, it is estimated that if the cost of the foreign packaged software is 1 then a localized and converted program is 2 and a semi-customized package is 5 to 6.

While initially localization presented some problems for foreign software developers, the process has now been refined through learning by experience and changes in the format most Western operating systems and application packages to accommodate double byte systems (AEA July 1992). Such conversion can now be done relatively quickly and easily, especially if it is an upgrade of an existing program that has already been localized. Several companies such as Apple, Microsoft, and Hewlett-Packard have modified the operating systems and software tools they supply to 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 shorter. In addition, since many new application programs are upgrades, the amount of new programming to be localized has been reduced, further facilitating the process.

To better serve customers, the major Japanese integrated systems producers such as NEC, Fujitsu and even IBM Japan have opened offices in the US to both attract and facilitate conversion of new programs to their operating systems. In this manner, Japanese integrated system producers are subsidizing and assisting the entry of the US software industry into the Japanese market, simultaneously lowering costs and expanding the user base for US firms.
Japanese customers and integrated producers are maintaining their own strategic integrity by improving foreign developers’ competitiveness. In the process, they have undermined MITI’s attempt to develop an independent, globally efficient, Japanese packaged software industry.

As expected from this analysis, the new, high growth software market segments in Japan are dominated by foreign packaged 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 (AEA July 1992). Novell and Oracle dominate network software. Nevertheless, customization persists, requiring localization and conversion along the spokes of the hub and spoke approach described earlier. Sun and HP are examples for firms that have structured themselves to do this. In the case of Sun, they are also receiving assistance from Fujitsu.

Customers’ desire for open systems to facilitate communication among the multiple platforms that have grown up within them or their company groups also fosters market penetration by foreign packaged software. Customers have adopted a multi-vendor purchasing system to control costs, give themselves more solution flexibility, and avoid excessive dependence on their mainframe suppliers. Also, for certain uses within a firm, particular hardware-software combinations are considered to be superior, e.g. UNIX and engineering workstations or data bases and IBM compatible mainframes. The interviews indicate that firms want to integrate such systems into their customized information networks.

At the same time, though such internal communication is considered very important, communication with other firms or networks is seen as less critical. This is because interfirm
communication is not as frequent as communication within the firm and within particular functions. So the fact that these other firms might use different platforms is not considered a current problem for the companies interviewed, especially as such interfaces conform to standard industry protocols. Therefore, 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 conforming to standard industry protocols between network servers (TCP/IP) for horizontal communication with outside parties. 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. Producers of RISC chips and related operating systems are better able to do this then competitors or other suppliers because of their intimate knowledge of their own MPUs’ capabilities.

**Game Systems**

Confirming the validity of the evolutionary approach over cultural or best practice theories as an explanation of Japan’s software industry is the fact that in games, Japanese firms do not suffer from these adverse economic conditions and are global leaders. For games producers successfully pursuing a global user base strategy, it would be natural to expect them to begin to pursue a hub and spoke strategy as well if the games market were to become fragmented, with multiple operating systems analogous to the situation in the business software
market. In fact this is now happening. When Nintendo had 70% of the global video game market, a software company could produce just for Nintendo and capture most of the global user base. However, the playing field changed. Nintendo lost share to Sega, and more powerful game machines were introduced by Sony, Sega and NEC, incorporating CD-ROM capabilities and more powerful MPUs. In addition, more and more PCs are being sold with multimedia capabilities, including a CD-ROM player and video chip. Now there is little difference between developing a sophisticated game for new game machines or for a PC, and the customer/machine market has become more fragmented with different operating systems. Under these circumstances, just as in the business market, a hub and spoke strategy makes the most sense to capture and maintain the largest user base. Using this strategy, a game software company develops a basic game program and then adopts it to different operating systems. Thus, one introduces a Nintendo, Sega, Sony, NEC, IBM/PC and MAC version of the same game, writing the programs to accommodate the different application interfaces. Several of the large game producers have already successfully begun to do this.

From this and similar examples, one can see that Japan’s failure to develop a strong business packaged software industry is due primarily to customer economics and strategies in combination with the industry’s evolution, not to management failure to adopt best practice or to cultural differences between the US and Japan. This situation also confirms Baba et al's conclusion (1993) that Japan's software industry is heavily customer driven. Sega and Nintendo have leveraged a large, rapidly growing domestic market for games to adapt to the global market while several of their major software developers are now pursuing a hub and spoke strategy, just
like their US business software counterparts, to deal with the emergence of a multiple platform market for games. 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. While the large household investment in game cartridges has acted as an entry barrier to other firms and a deterrent to users' switching among systems as long as the technology has remained stable, the emergence of CD-ROM and 64 bit processors as the de facto industry standard is changing the market. First-mover advantages during periods of technical upgrade appear to be important in this market, 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 US software producers' pricing new programs aggressively to get the subsequent upgrade revenues. The US packaged software and Japanese game industries thus share many common economic characteristics.

**GOVERNMENT POLICIES, OBJECTIVES AND INITIATIVES**

**Alternative Paradigms**

Nevertheless, in terms of US public and industry policy, there remains a need to monitor MITI’s new development strategies and to keep the market for US software in Japan open (Prestowitz, Morse and Tonelson 1991). US managers should continue to exploit the hub and spoke strategy while learning more about the market from Japanese systems producers and large software customers and their captive software subsidiaries. US 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 policies such as 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.

Historically, an important research topic has always been that successful Japanese industry international competition has often had a political and public policy dimension. These considerations have clearly extended to software. However, the US policy interest seems to be more in maintaining the US dominant position in packaged software than in protecting the US industry from an influx of Japanese imports since that has not occurred, though games have been an exception, as the US government has investigated Nintendo for possible attempts to monopolize the video game market. Nevertheless, MITI’s involvement in software development initiatives, past and present, and their support of Japan’s computer industry have logically extended the research on the software industry into the political economic sphere. The research team recognized Japanese MNCs have often been influenced in their process of domestic evolution, growth and international development by Japanese government policies and outside political economic pressures. (Anchordoguy 1995 paper expected based on participation in this research project.) This has been true in computers (Anchordoguy 1988 and 1989) as well as in major industries such as steel, autos, semiconductors, satellites and machine tools. Japanese trade, investment and industrial policies have been the topics of extended and often acrimonious negotiations with the US government. MITI from time to time has tried to change the regulations applying to software, such as trying to alter its intellectual property protection in 1984 or the treatment of decompilation in 1994.

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 much less of an issue, with the exception of negotiations over intellectual property rights. In part, this has been because major Japanese software customers have increased their independence from the government (Rapp 1993). When questioned about the government’s influence on their software decisions, all large customers responded “none”. Similarly, large customers said that MITI had no ability to force them to buy domestic software as the Ministry had done earlier 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.

COMPETITIVE DYNAMICS

Various explanations of Japan’s industrial competitiveness provide 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 the technical and organizational changes necessary to take maximum advantage of changes brought on by new software markets and technologies, such as network servers and open systems. This should occur even ahead of their Western counterparts. This has not happened. The evidence supporting the evolutionary hypothesis indicates that only certain software is favored by Japan’s competitive strengths. Those strengths
favor graphics, games, and certain multimedia applications. The tendency to emphasize customization and to form subsidiary software development companies favors intra-industry software advances over inter-industry expansion.

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). The software human resources have remained concentrated and in captive arrangements in user industries, such as autos and steel, which remain globally competitive. Thus, these resources have never been released for the development of an independent packaged software industry. At the same time, this analysis of the software industry benefits directly from 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). It also supports research indicating that corporate history and administrative heritage matter in corporate behavior and successful competition (e.g. Bartlett and Ghoshal 1989; Nelson 1991).

Another analytical approach to explaining 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. As noted earlier, the cultural disadvantage paradigm cannot explain this outcome and thus fails as an explanation what is happening in this industry.

**Evolutionary Approach**

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. (In current economic jargon, path dependency matters.) These rules and routines govern strategic and operational decisions and implementation which affect R&D, production, human resources, finance and market expansion. As such, these rules and routines 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 in games. The development of Japan’s software industry also offers an opportunity to test Nelson and Winter’s evolutionary theory of the firm (1982) to explain the industry’s weak global competitiveness other than in games. The research indicates as well that though Nelson and Winter focus on intra-organizational concepts, the theory can be used to analyze relations between institutions too, such as those between MITI and large Japanese firms that are major software customers or producers. This analytical approach is used to explain 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 US for the eventual substitution of computer networks for mainframes, and the continuation of large Japanese firms’ preference for customized rather than packaged software.
The analysis has emphasized the differences in customer 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 Novell, Oracle or Microsoft.

This intra-firm view has been used to explain Japanese corporate success in studies such as Baba (1989) and Rapp (1992 and 1993); it is now being used to explain failure to achieve results in a targeted industry. 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 practices, a hypothesis 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). When the firm, however, believes its stay-put decisions are rational and cost effective, achieving a change in direction without strong policy measures is almost impossible. This is the current situation with respect to large Japanese software customers.

Nelson and Winter argue that past Japanese competitive successes have resulted from logical adaptations to business, economic, and political circumstances over many years. Successful forms, policies and procedures developed over that time turned out to be extremely competitive in current world markets, at least for certain industries and products. This hypothesis of course 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 in certain industries is still real, it is
not necessarily as all-encompassing or as clearly transferable to different products or competitive environments (Florida and Kenney 1991). This view has in turn been borne out in the software case.

The Nelson-Winter analysis therefore differs from studies by analysts such as Womack, Jones, and Roos (1990), Krafcic (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 US software producers could 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 US software producers should--and can--copy this system, which is best practice, to become equally efficient and cost competitive in software. The preceding analysis describes how this has not occurred and explains why.

Another analytical approach to organizational competitiveness is the view that for cultural or institutional reasons, managers in leading Japanese firms are more capable than their US, 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 and costs 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 such as semiconductors. 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 develop a competitive domestic software industry--except in the government unsupported area of games (Rapp 1993A). Thus this argument fails as an explanation as well.

Of course some American policy makers and corporate executives claim the credit for implementing an astute global strategy that facilitates more rapid adoption of best practice by US firms while protecting software producers’ intellectual property rights and market access. This view is similar to the best practice hypothesis. Others say it is due to superior US cultural attributes with respect to software development, favoring US program developers over their Japanese counterparts (Delaney 1994). This proposition also fails because it is inconsistent with Japanese success in games. Recent developments in Japan’s software industry thus offer interesting opportunities to test the relative merits of various approaches in explaining Japanese competitive development in software. Evolutionary theory in turn has proved the most productive in explaining the research results.

It is the specific characteristics of the Japanese computer industry’s evolution that have been the primary constraint on customers’ ability to adopt software best practice or even superior developments such as the substitution of client (network) servers and workstations for mainframes. This has constrained Japanese suppliers from diverting programming resources to
these new technologies. The major integrated suppliers have maintained their strategy of separate mainframe platforms to hold customers more closely. They have then combined this approach with improved vertical compatibility between the mainframe and their client servers, workstations and PCs by using families of microprocessors (MPUs) in different hardware. These processors, while compatible with other operating systems, are optimized for the separate platform. This strategy has perpetuated industry fragmentation and made it more efficient for major computer manufacturers and large customers to adapt existing foreign packaged software than to try to develop totally new software on their own. In this manner, evolutionary analysis helps explain why earlier predictions of US competitive decline in software (Feigenbaum 1983, Cusumano 1991) have not materialized even as competition in high technology electronic hardware remains intense.

**COMPARISONS AND CONTRASTS WITH THE US**

In his excellent paper on the history and development of the US software industry, Edward Steinmueller present four features that he feels have determined the organization and structure of the US software industry (Steinmueller 1993). These are: the tradeoffs between company - user created and package software that impact make or buy decisions; the effect of organizational issues on the production and use of software; the use of software in networks that define entry points and the ultimate shape and scope of networks while pursuing the addition of
users to create external economies; the technical interaction between hardware and software improvements that affect the adoption of new hardware and software.

Examining these four dimensions, one sees important differences between the US and Japan. In Japan, the bias is towards make over buy decisions because of the perceptions of Japanese customers of the benefits of their unique software systems. This in turn is supported by the creation of numerous software subsidiaries by large customers, a phenomena that does not exist in the US. In turn, the primary purpose of networks in Japan is not to achieve external economies by expanding the user base but rather to extend control over the system and maximize its utilization by linking currently disparate systems together under a three tier management system. The entry points are thus well defined by the systems to be linked. Finally, there is currently very little reengineering going on in Japan, so that the real link is the need for new hardware to perform new tasks which create the demand for new software which must be customized to fit into the existing system.

In Japan and in the US employment policies and practices appear to be mirror images. In American firms EDP departments are generally run by people with strong EDP backgrounds who have their careers as EDP specialists rather than as generalist managers. This has been reinforced by US labor mobility, combined with the industry's pioneer status. Customers initially hired technical experts from systems developers and then from other large customers. Further, the development of systems integrators and software producers such as 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 established corporate context.

Since US EDP specialists in corporations often view themselves as computer specialists first and corporate executives second, they have looked to the computer industry to facilitate their mobility. When they have shifted from one firm to another, they have not wanted 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 in computer software. Universities that use computers and offer courses in computer science have been caught in the same push for standardization and career mobility. These institutional, economic and management forces have promoted further integration; in contrast these factors in Japan have encouraged fragmentation.

Given the technical expertise necessary to run large mainframe computers during the industry's early development, careers in the computer industry have been facilitated by customers and the lock-in effects to a particular standard technology and supplier, as technicians moved from customer to customer. Suppliers promoted this mobility, training and career development since it helped them secure their own markets longer term. At the same time, this pattern suggests why many corporate EDP departments were reluctant in the 1980s to embrace PCs until the introduction of networking. Many (MIS) Management Information Systems’ 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 to 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 essential to access large 
mainframe databases. This partly explains the origins of the CIO, Corporate Information Officer. 
In addition, the increase in computing power that made the PC revolution possible also made 
possible very large systems requiring very sophisticated expertise. Even in the US, there is a hard 
core of mission-critical and other large scale computing uses that will require mainframe 
capabilities into the foreseeable future.

In the US the availability of generic programming and of relatively inexpensive 
prepackaged solutions has meant organizations have had to do more to make this software work 
to service their specific needs. Under these circumstances, it is hardly surprising that the number 
of programmers and EDP expenditures by software customers in the US 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 client and 
customization is the norm, direct internal customer expenditures and personnel are smaller and 
are roughly equivalent to that for software systems houses. This situation is reinforced by the 
fact that software providers in Japan often rent programmers to firms under contract for specific 
projects (Baba et al 1993).

The reengineering phenomenon sweeping corporate America is in part an attempt to 
achieve a better integration of a firm's basic business, operating personnel, and information 
systems. Such integration already is the case for most large customers in Japan; yet Japanese 
government policy wants to move more towards packaged solutions. The US reengineering trend
is driven by the fact that many firms feel they have not achieved the expected productivity increases for their investments in information systems (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 the US EDP heritages noted above where purchase decisions were made for the convenience of the EDP specialists running these activities, not necessarily 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 has been able to significantly improve this integration through reengineering, the productivity results have frequently been amazing; this has forced other firms to respond. IBM's “solution” marketing is clearly a response to these pressures. And 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 for many years. 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 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. This is why reversal of this trend seems unlikely. The current joint ventures and cooperation between US software developers and Japanese customizers cater to large clients’ 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.

**ANALYTIC RESULTS**

Many leading Japanese manufacturers have out-competed US firms worldwide in a number of industries in which the US was once a leader, with adverse consequences for growth, employment and profits at the US 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 have supported various arguments on how these differences resulted 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), Baba (1989), Bartlett and Ghoshal (1989), Clark and Fujimoto (1991), Cusumano (1985, 1988, and 1991), Franko (1983), Harvard Business School (1988, 1989), Krafcik (1988), National Research Council (1992), Porter (1990), Rosenbloom and Cusumano (1987) and Smitka (1991).
Indeed, studies such as those by Womack, Jones, and Roos (1991) and Krafcik (1988), have argued that Japan's advantage is in process and manufacturing technology. They believe lean production systems can be applied with competitive advantage to any environment or industry. Yet, others like Dertouzos, Lester and Solow (1989) note that Japanese firms are not leaders in industries using continuous production 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 is not a continuous manufacturing process, but shares many of the discontinuities of semiconductor or automobile production. It can thus be organized similarly with similar results. The system described by him substantially reduces the time and cost of designing and producing a given amount of software code while improving quality and reliability. Over time, given higher Japanese growth 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 US 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 US firms.

Additionally, as in ICs, the Japanese government had 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 for software 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 the 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. 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, as explained above, this projected scenario failed because it did not recognize the difference in industry economics for software producers and customers from ICs.

**FUTURE EVOLUTION**

It is the difference in economic structure of software compared with many other products that has allowed many US firms to successfully enter, and even dominate, particular Japanese software market segments. Contributing to the software and computer hardware industry’s evolving hub and spoke structure is customers’ increasing preference for the three tier computer system whereby mainframes manage network servers that manage several workstations and PCs. Because of their huge investment in customized mainframe software operating on a highly differentiated operating system, large customers are not yet 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. Nonetheless, they want to move in this direction and to operate a multi-vendor system. Consequently, many have 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 the 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 in order to maintain market advantage in total systems and hardware. Foreign suppliers will use these motivations to increase their Japanese and global user base. In this way, the hub and spoke strategy pursued by the foreign software suppliers and fostered by the established computer hardware and operating systems suppliers will increase foreign firms’ already considerable competitive advantages in the Japanese market. This conclusion is confirmed by the survey results. This is why 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 customers through localization, conversion, and customization of foreign packaged software, and on developing additional applications for the existing systems. This industry structure thus relies on foreign development and a relatively stable client base where, except for games, there is little independent software R&D. Overall, the
entry and presence of foreign vendors in the Japanese market will continue to be supported by the large integrated systems producers, hurting the development of domestic Japanese software firms.

In sum, the presently fragmented operating systems and software application pattern will persist, and standardization will not occur except for some workstations. Except for games or software bundled in competitive manufacturing systems, internationally competitive packaged software is unlikely to be developed in Japan unless large clients’ desire for customization dramatically declines. But since there is no economic or management incentive for this to happen, given the customer firms’ high sunk costs and strong desire to support their unique competitive operating advantages, it does not appear a likely evolutionary direction. Rather, in the future the Japanese software industry will probably be composed primarily of profitable niche players, e.g. those producing Japanese language word processing and accounting programs, or those affiliated with foreign firms.

This analysis argues that predictions that the US 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 would seem to have undermined the competitiveness of its independent software companies. In addition to strong user interest in customization, Japanese management practices also adversely affect Japanese software producers’ competitive position. Since the first goal of a Japanese corporation is survival, major systems producers such as 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, in their willingness to facilitate the localization of foreign software to their operating systems and middleware (AEA 1992). Having secured their base businesses, the system producers look for growth opportunities in the same domestic market, such as workstations or PCs, in emerging markets abroad, like China, or in related markets, such as multimedia and games. They are likely to try to extend their software alliances to these new opportunities, again favoring the foreign producers and the expansion of their global user bases. "Competitive compulsion" will than force competitors to follow suit (Yoshino 1968; Ohmae 1991).

Another practice that has promoted fragmentation and thus hindered 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, in software the system favors customization over packaged software due to Japanese firms’ ability to train and retain EDP personnel. Conversely, the mobility of the American labor force has required greater standardization of both operating systems and application programs. In sum, many of the competitive advantages of Japanese firms and management identified by Aoki (1990) and Cusumano (1991) do not exist for software; it is not surprising that MITI’s efforts to extend its successful hardware paradigm to software have not worked. The failure of industry support for its policies is forcing MITI to look for a new approach. But it is not clear officials are taking sufficient account of user economics and customers’ business objectives, or the different structure of this industry compared to the
industries in which previous government promotion policies succeeded.

**SUMMARY AND CONCLUSIONS**

This report 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 predicts, this research demonstrates that even leading Japanese firms have some difficulty in adapting to new global competitive conditions. Rules and routines that produce success in some industries cannot necessarily be applied to others, notably the software industry, with the same success. Japan is quite competitive in a few software segments: graphics, games, and certain multimedia applications. However, in most 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 their production processes. Localization and adaptation dominate the corporate software market in Japan, especially for mainframes.

Evolutionary theory maintains certain corporate rules and routines develop in response to situations or events that often have little to do with their later benefit or disadvantage to the industry or firm. These practices then evolve over time in response to changes in economic, industry, and firm circumstances, bringing more change. Corporate history and administrative heritage matter. A key objective of this study has been to identify the common and unique elements in Japan’s software industry as well as in specific market segments that will affect the industry’s future evolution. The research has identified the emphasis on customized software,
driven by what firms see as their competitive advantage, as a critical industry characteristic, contributing to a high degree of fragmentation and a continuing strategic advantage for US firms.

Yet history indicates that Japanese firms and their competitive challenges are always moving targets. Despite favorable evolutionary developments to date, foreign software developers need to keep in mind the agendas of their Japanese customers and of the major systems integrators as they assess the unfolding competitive scenario in software in Japan. 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 both the industrialized countries and the newly industrialized Asian economies. So far, however, this effort has primarily involved adapting and customizing foreign software. Any change in this pattern should be a cause for concern for US software developers: if the preference for customization were to decline sharply, the market for software would change dramatically.

The key industries in signaling any such change 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 could possibly pull Japanese software into global competitiveness. 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 also accommodate possible changes in government policy and company strategies as MITI and leading corporate players respond to the lack of success in past
initiatives. Computers and software are global businesses 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. Excess reliance on cultural impediments to Japan’s emergence as a major global software competitor (Delaney 1994) is not realistic, as the case of games has demonstrated.

Nonetheless, the US industry has been quite proactive in the Japanese market. In dramatic contrast to the early stages of Japanese manufacturing development, they have established their own interactive and self supporting beneficial product demand cycle, similar to the Japanese consumer electronics industry’s support in terms of the demand for Japanese ICs. In the latter situation, the growth in Japanese consumer electronics increased the demand for ICs. This reduced the ICs’ cost according to experience curve economics, making the Japanese electronics that incorporated them lower cost and more competitive. This resulted in increased demand for those Japanese consumer electronic products and for more Japanese ICs, etc., etc. The comparable US example has been the increased use of workstations and personal computers, which increased the demand for MPUs where US producers have established a strong global position, including in Japan. As US PC and workstation manufacturers successfully penetrate the Japanese market by aggressive pricing to build market share (AEA July 1992), they create demand for US packaged and semi-customized software for workstations and PCs. This expansion in their user base makes their software more competitive, creating more demand for software and hardware. They are helped in this process by a strong yen and new US originated
operating systems that are more user friendly in the Japanese language. This development then reinforces the 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. The spillover from multimedia to PCs and workstations is unclear.

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 US software producers, which they are extending. For example, American personal computers typically have either DOS or MAC-based operating systems (and Windows builds on DOS to make it look more like MAC, indicating format 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 chips. Users' compatibility demands have forced commonality, further helping Intel and Motorola, as computer manufacturers have demanded their chips or chips compatible with their designs.

This has created 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 numerous alliances, 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, such US firms as 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, this situation seems little different from the one that failed in the late 60s and early 70s. The ministry still seems to be trying to treat software as an autonomous 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 planning. 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 appear to 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 customers make investments overseas. Otherwise the government’s options appear quite limited.
As long as software is technologically advancing and changing rapidly, US producers appear to be at an advantage. They can more easily maintain this pace in software because development is less capital intensive than ICs, and the cost of the primary input, skilled people, is a current cost expenditure. Thus a lower capital cost is less of a benefit to Japanese companies than in more capital intensive industries such as semiconductors. US venture capitalists have actively supported US software companies, partly because they perceive Japanese and other foreign competition as weak. Because capital is cheap in Japan and readily available to large firms, major Japanese computer firms can support a range of software firms related to their hardware platforms, including foreign producers. This has prevented a more rapid rationalization of the Japanese software industry.

Since there is no dominant global software design across all equipment and hardware segments, potential economies of scale from dominating a domestic Japanese market for such a design 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, domestic dominance built on strategies to perpetuate a unique market position based on using systems that are incompatible with any of the global standards has created problems for firms such as 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 since it would become too easy for customers to switch to other producers as they upgrade to new machines. Further, as their PCs can communicate with their larger
computers, such switching could affect these sales as well. Still, the two-operating-system strategy is like riding two racing horses simultaneously, hoping you can successfully jump to the right one at the last minute.

In sum, the presently fragmented operating systems and software application patterns in Japan will persist and standardization will not occur, except for some workstations as firms move towards UNIX and open standards. Further, except for games, locally developed, internationally competitive packaged software is unlikely to occur unless customers’ desire for continued customization changes for which there is no economic or management incentive. Integrated system suppliers also have an incentive to maintain the present system, which keeps users tied to them.

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 report predicts software segments favored by Japan’s competitive strengths will be graphics, games, and certain multimedia applications, but mostly will be the unique customized systems supporting competitive prowess in industries such as steel, autos and consumer electronics.

These manufacturers have invested aggressively in software and other productivity enhancing technologies to lower costs. So far, this has been primarily in either customized software or foreign packaged software adapted and customized to their needs. Any change in this
situation should be cause concern for foreign software producers. 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 in most market segments has grown more quickly than the US. Large firms or groups can still subsidize growth via leverage or the stock market, while capital availability remains an advantage even though low interest rates in the US and a hot IPO market for high technology stocks, especially software, have done much to eliminate this advantage. Failure to recognize the economic and organizational impediments due to the industry’s evolution would thus be a policy error. This understanding will hopefully help the US and Japan to better manage the realities of a global software market and continuing political friction surrounding their competition in high technology.

REFERENCES AND BIBLIOGRAPHY


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


118


APPENDIX I - SIMULATION ANALYSIS AND EXPLANATION OF SIMULATION

VARIABLES

Simulation

In order to further analyze the effects of the variables and issues that the research team sees as most critical to competition in the Japanese software market, the team is simulating the Japanese market for packaged computer software. The market is segmented into the demand for packaged software in mainframes, network servers, workstations and PCs. The game sector is not included, since its economic structure is quite different than that for business packaged software. The customized market influences the demand and supply functions as an alternative either to buying packaged software or buying a software package plus semi-customization. As the relative cost of customization increases, the demand for semi-customization and packaged solutions rises. The greater a customer’s installed customized base the greater its preference for semi-customization as opposed to a pure packaged solution.

The demand function has two parts. One is demand by users who are already using a particular software; they are influenced by their installed base plus price and the market share of the various suppliers. In the other case, new buyers are only influenced by price and the market shares of the software producers. Market shares influence the buy decision according to a probability function. The supply function for each segment is primarily driven by the size of a firm’s user base, with global and Japanese sales serving as a proxy for the size of the global and local user bases. For foreign firms, a 1, 2, 5 relationship between the cost of the foreign package, the localized package, and the semi-customized program is used to arrive at a price to the
customer, and the relationship between a change in the user base and the price in the market. An interesting aspect of this analysis is that the impact of a change in exchange rates acts powerfully at the package level but is heavily diluted as a cost factor as part for a semi-customized package. Experience curve effects are factored in as a reduction in development costs as particular program is improved or upgraded. The object of the simulation is to analyze the effect of changes in relevant variables over time on market share.

The research team is currently specifying these and other relationships identified by its research project as being important determinants of packaged software demand in Japan in order to simulate their effects on market share. These variables are outlined and discussed below. This simulation will allow the team to experiment with different strategy and policy variables. Moreover, potentially this simulation analysis can be compared to a similar simulation being developed for the computer industry by Nelson and Winter. However, as the variables and relationships are taken directly from the research results, it is not expected to alter any of the basic parameters, trends or important variables already identified in this report.

**Explanation of Simulation Variables**

1. **Simulation Variables Directly Affecting Competition for Japanese Software Market Share and the Global User Base**

**Marco Economic Variables**

*Economic Growth* - The nominal growth in the economy is related through an elasticity measure to the growth in the market for particular software, i.e. mainframe, PC, etc. on an historical basis over the last five to ten years. This relation can then be projected forward into the future. In this way the simulation tries to develop an elasticity of market growth projection based on the past elasticity of demand for different types of software versus projected economic growth as estimated by various economic forecasting models.
The Cost of Capital to Japanese competitors given the current stock market slump and the fact that most large players still have good access to loans takes the interest rate as a proxy. This is then used in the simulation to estimate how much it costs Japanese companies in terms of capital to develop a particular piece of software if they were to capitalize this expense. Labor costs in turn are gathered from industry statistics and then related to development costs. These costs are then impacted from a competitive standpoint by the yen exchange rate in terms of yen versus dollar which can also be taken from that used in various economic forecasting models.

**Industry Variables**

One important relationship that is estimated from existing data is that between installed computer systems (Super, Mainframe, Mini, Workstation, and PC) and the market for that type of software. Similarly, the growth in the installed computer base can be related to the growth in demand for that type of software. Finally, technological change in computer systems such as faster more powerful MPUs can be factored in later as a dummy variable influencing the demand for particular hardware and in turn spurts in demand for its associated software.

Competitors’ global user bases are estimated using their sales or current market shares as a proxy.

Competitors’ Japan user bases are estimated the same way using the data available on Japanese market share by computer type and company.

Competitors’ development costs for new products are initially assumed to be the same as one’s own development cost for a similar product.

Market growth (Super, Mainframe, Mini, Workstation, PC) is factored into the simulation as part of the elasticity of market growth related to economic growth as explained above.

Competitors’ growth is an outcome of their user base (current market share), the growth in the market and the effect of other variables.

Competitors’ relative prices are a function of their user base, assuming a constant mark-up.

The supply of skilled labor (programmers, software engineers, etc.) is taken from published industry data. Then from the questionnaires, industry data and software expenditures, the simulation can estimate how many of these are being used by large firms for customization (maintenance and new programs) and therefore how many might be available to produce packaged software. These skill supply can act as a constraint on the ability of a packaged software industry to develop. It can also indicate the potential savings in skilled labor to a firm of using semi-customized software solutions.

Localization/adaptation costs for foreign suppliers of new programs and for upgrades by foreign suppliers as developed from the questionnaires is related to software’s percent of total user costs per unit of output to get at supply costs and the supply function by software market.
User conversion costs, the externalities among multiple users of similar software and the number of incompatible operating systems or computer architectures are related to large users’ desire to use customize software at greater expense. The effect of the yen these factors is estimated.

Learning curve economies from repetition of programming similar software is used to estimate possible economies in customization using the multi-subsidiary or software factory approach. Costs usually drop 20-30% each time experience doubles. This cost reduction model is run against a user based economic model and one involving semi-customization. The simulation is used to estimate the possible benefits and costs of the two approaches, including the implicit costs and implied production productivity benefits to large firms who persist in using customized software.

Using the simulation is may be possible to compute a “piracy” variable which would estimate the cost of low priced pirated software to potential local competitors versus cost to the developers. On a default basis, this variable would be estimated from user base economics. As such it may be possible to show that the relative effect on the global competitor’s user base is less than on the local Japanese competitor. In this case, piracy can be part of the first mover barrier and argues that more forceful Intellectual Property protection is necessary to develop a domestic packaged software industry. Thus, the “piracy” variable is part of the simulation estimating the first mover cost barrier dependent on user base economics. In addition, it can be used to estimate the demand base barrier too, i.e. once a user is using a particular piece of software it may not see the need for another or a replacement, even if available on a pirated basis.

Internal or Firm Related Variables

New product development (niche), new product development costs and upgrade costs have been estimated from the market share and questionnaire data noted above. The default model in turn relates upgrades to the growth in sales.

The simulation model assumes that a firm’s own user base is indicated by current and future sales. The relative price that must be met to compete is then calculated by the firm’s need to compete with semi-customized or localized/adapted foreign software. Given a certain mark-up or after deducting the assumed cost of capital and labor (employment) based on industry data and the questionnaires, a firm’s cash flow and growth rate is a quantifiable outcome that can then be used to estimate its potential growth rate and increase in market share which will then affect its user base and costs on the next iteration.

The external user benefits and proprietary user benefits (the implicit value of tacit knowledge and of proprietary productivity improvements) are then quantifiable outcomes based on the actual cost to user of customization as gathered through the questionnaires versus the cost of using a localized package. Localization/Adaptation is also a cost related to packaged software which applies only to imports and for which data was obtained using the questionnaires. It indicates the cost/price target for Japanese developed packaged software which is used in the model. It also can be related to changes in the yen and the ability of Japanese software to
compete in the global market. The default model assumes production/reproduction costs are negligible.

Time to market as well as marketing and distribution costs are considered similar for like competitors. However, a foreign competitor with an existing product is considered to have an advantage over a local producer, while a foreign producer who has already localized/adapted a product would have a lead over those who have not. These competitive leads are available from industry data and the questionnaires.

2. Simulation Variables or Environmental Factors Indirectly Affecting the Competitive Environment but not simulated as direct inputs or outputs of the simulation

Macro Economic Variables

The permanent employment system affects labor mobility. It is related the opportunity cost of not switching to newer or packaged software. From that one may be able to assign a relative weight based on the results of the simulation

The tax system will affect costs, cash flow and growth rates but is not factored into the default simulation.

Industry Variables

Technological change in the development and production of software will change software costs but is not in itself an outcome of the competition for Japanese market share. Implicitly, the default model assumes that such change is widely diffused and benefits competitors equally.

Similar comments apply to changes in the technical knowledge base affecting computer hardware and software.

The cost to competitors of maintaining their software, i.e. constant improvement plus customer support and service, is assumed fairly uniform for competitive programs and thus is either neutral or subject to the same user base economics as development costs.

Various competitors’ upgrade costs are considered in the default case to be similar for like software returning to one to the user base model. If in fact they are different, this would have some impact on competitiveness, though it is only likely to be important when the user basis are nearly equal.

Government subsidies for this industry do not seem to have been effective because they operate directly on costs rather than on the user base. Therefore, in the default case, they are ignored or a considered to treat all competitors the same.

Users’ training costs (years of employment) will affect user’s cost of switching on abandoning customization. However, on the basis of the questionnaires, this appears to be similar for users in the same industry and thus does not favor one software house over another.
The availability of capital is generally a function of firm size or is usually readily available for large firms (integrated systems producers, software developers and users) but not for smaller firms. But since the study focused on large firms, it should not give any one a particular advantage.

University/educational support is an important variable affecting Japan’s ability to compete in the global market vis a vis the US and others. It can affect the cost of development, but the default model generally assumes it is covered in wage rates or is neutral across large firms.

Adaptation subsidies by integrated systems producers are not considered in the default case to give any foreign firm an advantage, and the pricing differential between foreign software and local or custom software already has the subsidy effect included.

The intellectual property system and its strong or weak enforcement (piracy) is assumed in the default case to apply to all competitors equally. If in fact this were not true as an aspect of government policy, its effect on costs and a firm’s user base would have to be estimated.

The effect of government support policies (regulations, protection and infrastructure) are captured in existing market shares and the current price/cost variables. As government policies are usually long-term, in the default case these are expected to persist. If there was a change, its affect on costs and various user bases would have to be estimated. For example, the recent decision to apply ISO 9000 standards to software would generally impact smaller software producers with fewer users more heavily. The actual effect could be estimated by looking at the costs of complying with ISO 9000 standards as a function of firm size.

Firm Related Variables

As stated in the text, the Hub & Spoke strategy is a way to expand one’s user base and thus is a strategic variable that can be changed to see its effect on market share longer term. In turn, acquisitions and strategic alliances are ways of affecting the user base and/or implementing an H&S strategy. However, as the default case implicitly incorporates an H&S strategy in the way the it works, the longer term effect of such strategic alliances or mergers on a firm’s cost structure could be estimated from the model by making one time adjustments in the firm’s or a competitor’s user base given the particulars of any given transaction or agreement.

R&D investment for similar products absent other data are considered similar in the default case for a packaged versus a custom program and as between competitors. Differences will thus primarily lie in the user base and/or the number of customized solutions done (i.e. learning effects).
APPENDIX II - JAPANESE QUESTIONNAIRES (ENGLISH)

COLUMBIA BUSINESS SCHOOL
RESEARCH PROJECT:
THE FUTURE EVOLUTION OF JAPANESE-US COMPETITION IN SOFTWARE:
POLICY CHALLENGES AND STRATEGIC PROSPECTS

Professor Bill Rapp, Principal Investigator
Columbia Business School
New York, NY 10027 USA

Phone (English) 604-721-7020
Phone (Japanese) 212-854-5936
Fax: (212) 678-6958

About this Questionnaire

The Japan - U.S. Friendship Commission, in collaboration with the Japan User’s Association for Information Systems (JUAS), the Center for International Business Education and Research at the University of Washington (CIBER), and the Center for the Study of Japanese Business at Columbia University, are jointly sponsoring a research project on the future of the Japanese software industry. We are interested in your company's strategy and future projection in the software industry.

We have worked hard to make the questionnaire easy to answer. We estimate it should take you only a few minutes to complete. To those participants who respond within 3 weeks, we will provide a detailed comparative study of both Japanese and U.S. software developers and users. This analysis will provide detailed information about important trends in needs and usage among major software users and developers.

Your responses to the questionnaire items will be kept in the strictest confidence. Your answers will be combined with the responses of other respondents to create an aggregated data sample. Our analyses will be conducted only on this aggregated sample, ensuring the confidentiality of your responses. In return for your participation, you will receive complete feedback on all our findings, including strategic and competitor information in your industry.

The items in this questionnaire cover a broad range of topics. Although you may not be well-versed on each topic area, please answer each item to the best of your ability. If you are completely unable to answer a particular question, please mark “NA”.

Thank you for taking time to answer this questionnaire.
Software Industry Questionnaire for Integrated Systems Suppliers

Name of Company: _________________________ Date: _________________________

Address: __________________________________ Tel: _________________________
__________________________________ Fax: _________________________
__________________________________

Contact person: ___________________________

1. Is your firm independent or an affiliate of a larger company?

2. If an affiliate, what is the name of the affiliated company or companies?

3. Is your parent or partner a producer of both computer software and hardware or only hardware?

4. How many employees in Japan are working on software development?

5. Does your firm own shares in captive Japanese software development companies?

6. If yes, what is their size in terms of sales and staff?
   sales?: ______
   staff?: ______

7. What percentage of your software was:
   developed by yourselves? ______
   by related companies? ______
   subcontracted to others? ______

8. When you contract outside for software, how is the supplier selected?

9. Do independent suppliers contact you to understand platform conversion, customization, and compatibility issues with respect to the software they will develop for you? your customers? their other customers?

10. With which systems or software operating platforms can your system easily connect or interface for interoperability?

11. Do your customers' different operations, functions, or groups sometimes use different platforms or systems?

12. The table below shows five different computer categories. Please tell us the percent of your firm’s total sales that come from hardware and software sales for each category. For example, if %10 of your firm’s sales come supercomputer hardware sales, put a “10” in the first box. If your firm has no sales in a particular case, please put a “0” in that box.

<table>
<thead>
<tr>
<th>Supercomputers</th>
<th>Mainframe or Minicomputers</th>
<th>Network Servers</th>
<th>Workstations</th>
<th>Personal Computers</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
12.b Now, for your **Mainframe or minicomputer customers only**, please tell us what percent of sales of the different computer types are to those mainframe customers. If you have no mainframe or minicomputer customers, please leave this item blank.

<table>
<thead>
<tr>
<th></th>
<th>Supercomputers</th>
<th>Mainframe or Minicomputers</th>
<th>Network Servers</th>
<th>Workstations</th>
<th>Personal Computers</th>
</tr>
</thead>
<tbody>
<tr>
<td>% sales</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

13. Now, for each of the five categories of computers, please help us understand what factors are important when your customers are making system selection decisions. Simply put an “X” in each box to show which factors your customers use.

<table>
<thead>
<tr>
<th></th>
<th>Supercomputers</th>
<th>Mainframe or Minicomputers</th>
<th>Network Servers</th>
<th>Workstations</th>
<th>Personal Computers</th>
</tr>
</thead>
<tbody>
<tr>
<td>customer’s historical usage</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>customers’ current operating platform</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>latest technology</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>price</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>reliability</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

14. Now we would like to find out how your competitors compete with you. For each (software) category, simply put an “X” in the box to show which factors your competitors use to compete against you.

<table>
<thead>
<tr>
<th></th>
<th>Supercomputer software</th>
<th>Mainframe or Minicomputer software</th>
<th>Network Server software</th>
<th>Workstation software</th>
<th>Personal Computer software</th>
</tr>
</thead>
<tbody>
<tr>
<td>interoperability</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>operating speed</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>rapid deployment</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>price</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>reliability</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

15. Do you supply application programs in addition to operating systems and middleware?

16. Please list the major software products which you supply for your customer

17. For each major software product, please describe distribution channels used. Simply put an “X” for all those that apply.

<table>
<thead>
<tr>
<th></th>
<th>Supercomputers software</th>
<th>Mainframe or Minicomputer software</th>
<th>Network Server software</th>
<th>Workstation software</th>
<th>Personal Computer software</th>
</tr>
</thead>
<tbody>
<tr>
<td>Direct marketing by you or a subsidiary</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
18. Now help us understand where your customers get their software. Use the chart below to tell us what percent of their total software purchases came from:
   1) your company or subsidiaries related to you
   2) their own in house software developers or their captive software suppliers?
   3) some third party software developers or integrated systems producers?

<table>
<thead>
<tr>
<th></th>
<th>Mainframe Customers</th>
<th>Non-Mainframe Customers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Your company or your subsidiaries</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Their own in house developers or captive software suppliers</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Competitors or integrated systems producers</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

19. Now considering your mainframe clients and your non-mainframe customers, how much of their software purchases in each of the five categories listed above came from:
   1) your company or subsidiaries related to you
   2) their own in house software developers or their captive software suppliers?
   3) some third party software developers or integrated systems producers?

20.a For your mainframe clients, what percent of your total sales to them is:
   1) developed in house or through one of your subsidiaries (%= _____)?
   2) subcontracted with an independent custom developer (%= _____)?
   3) converted and adapted packaged software from independent software vendors (%= _____)?

20.b For your non-mainframe clients, what percent of your total sales to them is:
   1) developed in house or through one of your subsidiaries (%= _____)?
   2) subcontracted with an independent custom developer (%= _____)?
   3) converted and adapted packaged software from independent software vendors (%= _____)?

21. Now show us which industries represent at least 10% of your software sales? Simply circle those industries which make up at least 10% of your firm’s software sales.
   Banking ... Securities ... Steel ... Retailing ... Automobiles ... Publishing ... OTHERS ________?
   (please write in others)

22. Considering your software subcontractors, which industries represent at least 10% of their software sales?
   Banking ... Securities ... Steel ... Retailing ... Automobiles ... Publishing ... OTHERS?

23. Considering the packaged software you adapt, which industries represent at least 10% of the supplier of packaged software sales?
Industry: Banking ... Securities ... Steel ... Retailing ... Automobiles ... Publishing ... OTHERS?

Function: Accounting ... Engineering ... Data Bases ... Word-processing ... Spread Sheet... OTHERS?

24.a If packaged software cost 10, how much would it cost to adapt to your use?
   10  20  30  40  50  More

24.b If customized software cost 10, how much would adapted packaged software cost?
   10  20  30  40  50  More

25. What percentage of your total software sales is application software?

26. What percentage of your total sales of application software is to:
   Users purchasing packaged or standardized software? _______
   Users requiring customization? ______

27. What percentage of your total operating or middleware systems is to:
   Users purchasing packaged or standardized software? _______
   Users requiring customization (please explain type and extent of customization) ______

28. Are the managers of the different EDP functions at your customer’s firm career EDP personnel, or do they come from other parts of the customers' company for a few years and are then rotated to non-EDP functions? Circle the option which best describes your customers.
   Career EDP personnel............Rotated to non-EDP functions.

29. What has been the effect of the government's push for open UNIX based systems and downsizing in the form of workstations and networks on your product or software development planning or operations?
   On your customers' EDP planning or operations?

30. To what degree will client servers and office computers combined with smart terminals replace mainframes in the Japanese market over the next five years?

31. In what industries and operations are mainframe or super computer use likely to continue indefinitely?

ALL INFORMATION CONTAINED IN THIS QUESTIONNAIRE WILL REMAIN CONFIDENTIAL.

Thank you for your kind cooperation in filling out this questionnaire.
Software Industry Questionnaire for Systems Integrators, Software Developers, etc.

Name of Company: _________________________ Date: ________________

Address: __________________________________ Tel: _________________
__________________________________ Fax:_________________
__________________________________
Contact: __________________________________

1. Is your firm independent or a subsidiary of a larger company?

2. If a subsidiary, name of parent and whether parent is a producer of both computer software and hardware, a user, or a software only producer?

3. How many employees are doing software development?

4. Does your firm own stock in captive Japanese software development companies?

5. If yes, what is their size in terms of sales and staff?

6. What percentage of your software was developed by such related companies and what percentage was subcontracted to others?

7. When you contract outside for software, how is the supplier selected?

8. Do independent suppliers usually have regular or formal contact with the integrated systems producers in order to understand platform conversion, customization, and compatibility issues with respect to the software they will provide their customers?

9. What % of customers use multiple platforms or systems?

10. With which systems or software operating platforms do you work or interface?

11. How are system selection decisions made by your customers (e.g. historical usage, current operating platform, latest technology, price reliability)? Do you supply software functions for mainframe? network? PCs/Workstations?

12. Please help us fill in the table below: The two columns represent one, your current mainframe or minicomputer software customers and two, your other software customers. Now for each cell please estimate the percent of your total software sales respectively for each category:

<table>
<thead>
<tr>
<th>Category I - Client Servers</th>
<th>Current Mainframe or Minicomputer software customers</th>
<th>Current other software customers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Client Servers</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
13. Respectively, what percentage of your major customers’ software expenditures are for super computers, mainframe, workstation, personnel computers, internal telecommunications or networking, and external telecommunications or networking?

<table>
<thead>
<tr>
<th></th>
<th>Super-computers</th>
<th>Mainframe computers</th>
<th>Workstations personal computers</th>
<th>int. telecom or networking</th>
<th>ext. telecom or networking</th>
</tr>
</thead>
<tbody>
<tr>
<td>Percent of major customers software expenditures?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

14. Do you supply application programs in addition to operating systems and middleware?

15. On cost basis, what percentage of application functions are developed by your customers internally?

16. In each software category, do your competitors compete primarily on price, reliability, rapid deployment, technology, interoperability, operating speed, other? Circle those which apply:

Price......reliability......rapid deployment......technology......interoperability......operating speed....Other?

17. Please list your major software products.

18. For each major product, please circle the main distribution channels used:
   1) Direct marketing by you or a subsidiary
   2) Direct marketing by independent systems integrators
   3) Independent outlets selling to large users
   4) Independent outlets selling to small and medium size business users
   5) Retail outlets selling to individuals

19. Now considering two types of clients described in the table below, what percent of their total software came from:
   1) your company or related subsidiaries
   2) their own in house software developers or captive software suppliers?
   3) some other software developers or integrated systems producers?

<table>
<thead>
<tr>
<th></th>
<th>Current Mainframe or Minicomputer software customers</th>
<th>Current other software customers</th>
</tr>
</thead>
<tbody>
<tr>
<td>your company or subsidiary</td>
<td></td>
<td></td>
</tr>
<tr>
<td>in house or captive supplier</td>
<td></td>
<td></td>
</tr>
<tr>
<td>other developer or integrated systems producers</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

20. For these two types of clients, how much of the software you supply them, do you
   1) develop in house or through a subsidiary?
   2) subcontract with an independent custom developer
   3) convert and adapt packaged software from independent software vendors?
21. Considering your own software production, please circle those industries which represent at least 10% of software sales. If there are OTHERS, please specify.

Banking ... Securities ... Steel ... Retailing ... Automobiles ... Publishing ... OTHERS?

22. Considering your software subcontractors, please circle those industries which represent at least 10% of their software sales. If there are OTHERS, please specify.

Banking ... Securities ... Steel ... Retailing ... Automobiles ... Publishing ... OTHERS?

23. Considering the packaged software you adapt, please circle those industries which represent at least 10% of their software sales. If OTHERS, please specify.

Banking ... Securities ... Steel ... Retailing ... Automobiles ... Publishing ... OTHERS?

Accounting ... Engineering ... Data Bases ... Word-processing ... Spread Sheet ... OTHERS?

24. (a) If packaged software cost 10, how much would it cost to adapt to your use?

10  20  30  40  50  More

24. (b) If customized software cost 100, how much would adapted packaged software cost?

50  60  70  More

25. Are the managers of the different EDP functions at your customer's firm career EDP personnel, or do they come from other parts of the customers' company for a few years and are then rotated to non-EDP functions? Circle the option which best describes your customers.

Career EDP personnel............Rotated to non-EDP functions.

26. What has been the effect of the government's push for open UNIX based systems and downsizing in the form of workstations and networks on your planning or operations?

...on your customers' EDP planning or operations?

27. To what degree will client servers and office computers combined with work stations and PCs replace mainframes in the Japanese market in five years? In ten years?

% of mainframes replaced in five years:

% of mainframes replaced in ten years:
28. (a) In what industries is mainframe or super computer use likely to continue indefinitely?

28. (b) In what functions is mainframe or super computer use likely to continue indefinitely?

29. Do you have any other views comments on the direction or structure of the software industry?

ALL INFORMATION CONTAINED IN THIS QUESTIONNAIRE WILL REMAIN CONFIDENTIAL.

Thank you for your kind cooperation in filling out this questionnaire.
Software Industry Questionnaire for User Companies

Name of Company: ________________________  Date: ________________  
Address: __________________________________  Tel: ________________
__________________________________  Fax: ________________
__________________________________
Contact: ________________________________

1. Is your firm independent or a subsidiary of a larger company?

2. If a subsidiary, does your parent have other affiliated software users or software producers?

3. How many employees are doing software development?

4. For your firm in Japan over each of the last two years what percentage of your total costs were represented by costs related to EDP, such as hardware, personnel, contract employees, software, and EDP consulting services?

5. Out of your Japanese EDP costs, how much was related to software developed by you for your own use?

6. What percentage of the software was purchased from non related companies?

7. How is your EDP operation in Japan structured and how many employees do you have working in each EDP function?

8. Does your firm have its own captive Japanese software suppliers?

9. What is their size in terms of sales and staff?

10. What percentage of your software was purchased from such related companies and what percentage from others?

11. What percentage of their sales are to you or your affiliates?

12. What percentage of software bought from others was represented by equipment manufacturers or their affiliates and what percentage by independent software contractors or systems integrators?

13. With how many systems or software operating platforms do you operate? Do they vary by function?

14. With which systems or software operating platforms do you work or interface?

15. Have you ever had to adopt a different platform or system in order to acquire or use a particular software package?

16. (a) How do you plan for changes in your hardware and software systems for each of your major EDP functions?

16. (b) When you contract outside for software, how is supplier selected?
17. How are system selection decisions made by your subsidiaries of affiliates (e.g. historical usage, current operating platform, latest technology, price reliability)? Do you supply them software functions for mainframe? Network? PCs/Workstations?
Mainframe:
Network:
PCs/Workstations:

18. Please help us fill in the table below: The two columns represent one, your current mainframe or minicomputer software supplier and two, your other software suppliers; Now for each cell please estimate the percent of your total software purchases respectively for each category:
Category I - Network Servers
Category II - Workstations
Category III - PCs

<table>
<thead>
<tr>
<th>Category</th>
<th>Current Mainframe or Minicomputer software suppliers</th>
<th>Current other software suppliers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Client Servers</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Workstations</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Personal Computers</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

19. Respectively, what percentage of your software expenditures are for super computers, mainframe, workstation, personnel computers, internal telecommunications or networking, and external telecommunications or networking?

<table>
<thead>
<tr>
<th>Category</th>
<th>Super-computers</th>
<th>Mainframe computers</th>
<th>Workstations</th>
<th>Personal computers</th>
<th>Int. telecom or networking</th>
<th>Ext. telecom or networking</th>
</tr>
</thead>
<tbody>
<tr>
<td>Percent of your software expenditures?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

20. What percentage of application functions are developed by you internally or through software developer affiliates?

21. In each software category, primarily on what do your vendors or vendors’ competitors compete?
Price     Reliability     Rapid deployment     Technology     Interoperability     Operating speed     Other?

22. (a) Please list major software products you purchase.

22. (b) Please list major software products you develop.

23. For each major product purchased, please describe distribution channel used to acquire software.
   1) Large Integrated Producers  
   2) Independent systems integrators  
   3) Independent outlets to large users  
   4) Independent outlets to small and medium size business users  
   5) Retail outlets to individuals

24. Now considering two types of suppliers mentioned in the table below, what percent of your total software in the last two years was developed by:
   1) your company or related subsidiaries  
   2) your systems suppliers’ own software developers or captive software suppliers?
3) some other software developers or integrated systems producers?

<table>
<thead>
<tr>
<th></th>
<th>Current Mainframe or Minicomputer software suppliers</th>
<th>Current other software suppliers</th>
</tr>
</thead>
<tbody>
<tr>
<td>your company or subsidiary</td>
<td></td>
<td></td>
</tr>
<tr>
<td>system suppliers’ own or captive software suppliers</td>
<td></td>
<td></td>
</tr>
<tr>
<td>other developer or integrated systems producers</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

25. Considering your own software development, please circle those industries in which you are an expert? If OTHERS, please specify.

Banking ... Securities ... Steel ... Retailing ... Automobiles ... Publishing ... OTHERS

26. Considering your software subcontractors, please circle those industries in which they are an expert? If OTHERS, please specify.

Banking ... Securities ... Steel ... Retailing ... Automobiles ... Publishing ... OTHERS

27. Please circle those industries and functions in which the supplier of packaged software is an expert. If OTHERS, please specify.

Industries: Banking ... Securities ... Steel ... Retailing ... Automobiles ... Publishing ... OTHERS

Functions: Accounting ... Engineering ... Data Bases ... Word-processing ... Spread Sheet ... OTHERS

28. (a) If packaged software cost 10, how much would it cost to adapt to your use?

10 20 30 40 50 More

28. (b) If customized software cost 100, how much would adapted packaged software cost?

50 60 70 More

29. What percentage of your total purchases of software is for:

Packaged
Customized

30. Are the managers of the different EDP functions at your customer’s firm career EDP personnel, or do they come from other parts of the customers' company for a few years and are then rotated to non-EDP functions? Circle the option which best describes your customers.

Career EDP personnel.............Rotated to non-EDP functions.

31. What has been the effect of the government's push for open UNIX based systems and downsizing in the form of workstations and networks on your EDP planning or operations?

32. To what degree will client servers and office computers combined with work stations and PCs replace your mainframes in five years? In ten years?

Five years:
Ten years:

33. In what functions will mainframe or super computer use likely continue indefinitely?

34. Do you have any views or comments on the direction or structure of the software industry?

ALL INFORMATION CONTAINED IN THIS QUESTIONNAIRE WILL REMAIN CONFIDENTIAL.

Thank you for your kind cooperation in filling out this questionnaire.
APPENDIX III - JAPANESE QUESTIONNAIRES (JAPANESE)
APPENDIX IV - US QUESTIONNAIRES

COLUMBIA BUSINESS SCHOOL
RESEARCH PROJECT:

THE FUTURE EVOLUTION OF JAPANESE - US COMPETITION IN SOFTWARE:
POLICY CHALLENGES AND STRATEGIC PROSPECTS

Professor Bill Rapp, Principal Investigator
Columbia Business School
New York, NY 10027 USA

Phone (English) 604-721-7020
Phone (Japanese) 212-854-5936
Fax: (212) 678-6958

About this Questionnaire

The Japan - U.S. Friendship Commission, in collaboration with the Japan User’s Association for Information Systems (JUAS), the Center for International Business Education and Research at the University of Washington (CIBER), and the Center for the Study of Japanese Business at Columbia University, are jointly sponsoring a research project on the future of the US and Japanese software industries. We are interested in your company's strategy and future projection of the US software industry.

We have worked hard to make the questionnaire easy to answer. We estimate it should take you only a few minutes to complete. To those participants who respond within 3 weeks, we will provide a detailed comparative study of both Japanese and U.S. software developers and users. This analysis will provide detailed information about important trends in needs and usage among major software users and developers.

Your responses to the questionnaire items will be kept in the strictest confidence. Your answers will be combined with the responses of other respondents to create an aggregated data sample. Our analyses will be conducted only on this aggregated sample, ensuring the confidentiality of your responses. In return for your participation, you will receive complete feedback on all our findings, including strategic and competitor information in your industry.

The items in this questionnaire cover a broad range of topics. Although you may not be well-versed on each topic area, please answer each item to the best of your ability. If you are completely unable to answer a particular question, please mark “NA”.

Thank you for taking time to answer this questionnaire.
Software Industry Questionnaire for Integrated Systems Suppliers

Name of Company: _________________________ Date: _________________________
Address: __________________________________ Tel: _________________________
__________________________________ Fax: _________________________
__________________________________
Contact person: ___________________________

1. Is your firm independent or an affiliate of a larger company?

2. If an affiliate, what is the name of the affiliated company or companies?

3. Is your parent or partner a producer of both computer software and hardware or only hardware?

4. How many employees in the US are working on software development?

5. What percentage of your total software sales are US sales?

6. Does your firm own shares in captive US software development companies?

7. If yes, what is their size in terms of sales and staff?
   sales?: ______
   staff?: ______

8. What percentage of your software was:
   developed by yourselves? ______
   by related companies? ______
   subcontracted to others? ______

9. When you contract outside for software, how is the supplier selected?

10. Do independent suppliers contact you to understand platform conversion, customization, and compatibility issues with respect to the software they will develop for you? your customers? their other customers?

11. With which systems or software operating platforms can your system easily connect or interface for interoperability?

12. Do your US customers' different operations, functions, or groups sometimes use different platforms or systems?

13.a The table below shows five different categories of computers. Please tell us the percent of your firm’s total US sales that come from hardware and software sales for each category. For example, if %10 of your firm’s US sales come from selling supercomputer hardware, put a “10” in the first box. If your firm has no sales in a particular case, please put a “0” in that box.

<table>
<thead>
<tr>
<th>Category</th>
<th>Supercomputers</th>
<th>Mainframe or Minicomputers</th>
<th>Network Servers</th>
<th>Workstations</th>
<th>Personal Computers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Percent of Sales</td>
<td>%10</td>
<td>%20</td>
<td>%30</td>
<td>%40</td>
<td>%50</td>
</tr>
</tbody>
</table>
13.b Now, for your US **Mainframe or minicomputer customers only**, please tell us what percent of sales of the different computer types are to those mainframe customers. If you have no mainframe or minicomputer customers, please leave this item blank.

<table>
<thead>
<tr>
<th>% sales</th>
<th>Supercomputers</th>
<th>Mainframe or Minicomputers</th>
<th>Network Servers</th>
<th>Workstations</th>
<th>Personal Computers</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

14. For each of the five computer categories, please help us understand what factors are important when your US customers make system selection decisions. Simply put “X” in each box to show factors your customers use.

<table>
<thead>
<tr>
<th></th>
<th>Supercomputers</th>
<th>Mainframe or Minicomputers</th>
<th>Network Servers</th>
<th>Workstations</th>
<th>Personal Computers</th>
</tr>
</thead>
<tbody>
<tr>
<td>customer’s historical usage</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>customer’s current operating platform</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>latest technology</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>price</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>reliability</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

15. Now we would like to find out how your competitors compete with you. For each software category, simply put an “X” in the box to show us which factors your customers use to compete against you.

<table>
<thead>
<tr>
<th></th>
<th>Supercomputer software</th>
<th>Mainframe or Minicomputer software</th>
<th>Network Server software</th>
<th>Workstation software</th>
<th>Personal Computer software</th>
</tr>
</thead>
<tbody>
<tr>
<td>interoperability</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>operating speed</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>rapid deployment</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>price</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>reliability</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

16. Do you supply application programs in addition to operating systems and middleware?

17. Please list the major software products which you supply to your US customers.

18. For each major software product, please describe US distribution used. Simply put “X” for all that apply.

<table>
<thead>
<tr>
<th>Distribution Type</th>
<th>Supercomputer software</th>
<th>Mainframe or Minicomputer software</th>
<th>Network Server software</th>
<th>Workstation software</th>
<th>Personal Computer software</th>
</tr>
</thead>
<tbody>
<tr>
<td>Direct marketing by you or a subsidiary</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sales through independent systems integrators</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
19. Now help us understand where your US customers get their software. Use the chart below to tell us what percent of their total software purchases came from:

1) your company or subsidiaries related to you
2) their own in house software developers or their captive software suppliers?
3) some third party software developers or integrated systems producers?

<table>
<thead>
<tr>
<th>Sales through independent distributor to large users</th>
<th>Mainframe Customers</th>
<th>Non-Mainframe Customers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sales through independent outlets to small and medium sized business users</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sales through retail outlets to individuals</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

20. Now considering your US mainframe clients and your non-mainframe US customers, how much of their software purchases in each of the five categories listed above came from:

1) your company or subsidiaries related to you
2) their own in house software developers or their captive software suppliers?
3) some third party software developers or integrated systems producers?

21. A For your mainframe US clients, what percent of your total sales to them is:

1) developed in house or through one of your subsidiaries (% = ____)?
2) subcontracted with an independent custom developer (% = ____)?
3) converted and adapted packaged software from independent software vendors (% = ____)?

21.b For your non-mainframe US clients, what percent of your total sales to them is:

1) developed in house or through one of your subsidiaries (% = ____)?
2) subcontracted with an independent custom developer (% = ____)?
3) converted and adapted packaged software from independent software vendors (% = ____)?

22. Now show us which industries represent at least 10% of your US software sales? Simply circle those industries which make up at least 10% of your firm’s software sales.

Banking ... Securities ... Steel ... Retailing ... Automobiles ... Publishing ... OTHERS? ____________?

23. Considering your US software subcontractors, which industries represent at least 10% of their software sales?

Banking ... Securities ... Steel ... Retailing ... Automobiles ... Publishing ... OTHERS?

24. Considering the packaged software you adapt, which industries represent at least 10% of the supplier of packaged software’s US sales?

**Industry:** Banking ... Securities ... Steel ... Retailing ... Automobiles ... Publishing ... OTHERS?

**Function:** Accounting ... Engineering ... Data Bases ... Word-processing ... Spread Sheet. OTHERS?

25.a If packaged software costs 10, how much would it cost to adapt to your use?
25.b If customized software costs 100, how much would adapted packaged software cost?

26. What percentage of your total US software sales is application software?

27. What percentage of your total US sales of application software is to:
   Users purchasing packaged or standardized software? ______
   Users requiring customization? ______

28. What percentage of your total operating or middleware systems is sold to:
   Users purchasing packaged or standardized software? ______
   Users requiring customization (please explain type and extent of customization) ______

29. Are the managers of the different EDP functions at your major customers’ firm career EDP personnel, or do they come from other parts of the customers’ company for a few years and are then rotated to non-EDP functions? Circle the option which best describes your customers.
   Career EDP personnel..............Rotated to non-EDP functions.

30. What impact will the US government's role in the Information Highway have on your: EDP Planning?
   Software Procurement or Development?
   On your customers' EDP planning and operations or software procurement and development?
   Other?

31. To what degree will client servers and office computers combined with smart terminals replace mainframes in the US market over the next five years? Ten years?

32. In what industries and operations are mainframe or super computer use in the US likely to continue indefinitely?

ALL INFORMATION CONTAINED IN THIS QUESTIONNAIRE WILL REMAIN CONFIDENTIAL.

Thank you for your kind cooperation in filling out this questionnaire. Please fax to (212) 678-6958
Software Industry Questionnaire for Systems Integrators, Software Developers, etc.

Name of Company: _________________________                                    Date: ________________
Address: __________________________________                                    Tel: _________________
__________________________________                                   Fax: _________________
[72x601]__________________________________
Contact: __________________________________

1. Is your firm independent or a subsidiary of a larger company?

2. If a subsidiary, name of parent and whether parent is a producer of both computer software and hardware, a user, or a software only producer?

3. How many employees are doing software development?

4. What percentage of your total software sales are US sales?

5. Does your firm own stock in captive US software development companies?

6. If yes, what is their size in terms of sales and staff?

7. What percentage of your US software was developed by such related companies and what percentage was subcontracted to others?

8. When you contract outside for software, how is the supplier selected?

9. Do independent suppliers usually have regular or formal contact with the integrated systems producers in order to understand platform conversion, customization, and compatibility issues with respect to the software they will provide their customers?

10. What % of US customers use multiple platforms or systems?

11. With which systems or software operating platforms do you work or interface?

12. How are system selection decisions made by your US customers (e.g. historical usage, current operating platform, latest technology, price reliability)? Do you supply software functions for mainframe? Network servers? PCs/Workstations?

13. Please help us fill in the table below: The two columns represent one, your current US mainframe or minicomputer software customers and two, your other US software customers. Now for each cell please estimate the percent of your total US software sales respectively for each category:

<table>
<thead>
<tr>
<th>Category I - Client Servers</th>
<th>Current Mainframe or Minicomputer software customers</th>
<th>Category III - PCs</th>
<th>Current other software customers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Client Servers</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
14. Respectively, what percentage of your major US customers' software expenditures are for super computers, mainframe, workstation, personnel computers, internal telecommunications or networking, and external telecommunications or networking?

<table>
<thead>
<tr>
<th>Percent of major customers software expenditures?</th>
<th>Super-computers</th>
<th>Mainframe computers</th>
<th>Workstations</th>
<th>Personal computers</th>
<th>internal telecom or networking</th>
<th>ext. telecom or networking</th>
</tr>
</thead>
</table>

15. Do you supply application programs in addition to operating systems and middleware?

16. On cost basis, what percentage of application functions are developed internally by your customers in the US?

17. In each software category, do your competitors compete primarily on price, reliability, rapid deployment, technology, interoperability, operating speed, other (please specify)? Circle those which apply:

- Price
- Reliability
- Rapid deployment
- Technology
- Interoperability
- Operating speed
- Other?

18. Please list your major software products.

19. For each major product, please circle the main distribution channels used:

   1) Direct marketing by you or a subsidiary
   2) Direct marketing by independent systems integrators
   3) Independent outlets selling to large users
   4) Independent outlets selling to small and medium size business users
   5) Retail outlets selling to individuals

20. Now considering two types of clients described in table below, what percent of their total software came from:

   1) your company or related subsidiaries
   2) their own in house software developers or captive software suppliers?
   3) some other software developers or integrated systems producers?

<table>
<thead>
<tr>
<th>Current Mainframe or Minicomputer software customers</th>
<th>Current other software customers</th>
</tr>
</thead>
<tbody>
<tr>
<td>your company or subsidiary</td>
<td></td>
</tr>
<tr>
<td>in house or captive supplier</td>
<td></td>
</tr>
<tr>
<td>other developer or integrated systems producers</td>
<td></td>
</tr>
</tbody>
</table>

21. For these two types of clients in the US, how much of the software you supply them, do you

   1) develop in house or through a subsidiary?
   2) subcontract with an independent custom developer
   3) convert and adapt packaged software from independent software vendors?

<table>
<thead>
<tr>
<th>Current Mainframe or Minicomputer software customers</th>
<th>Current other software customers</th>
</tr>
</thead>
<tbody>
<tr>
<td>develop in house or through independent subsidiary</td>
<td></td>
</tr>
</tbody>
</table>
22. Considering your own software production, please circle those industries which represent at least 10% of software sales. If there are OTHERS, please specify.

Banking ... Securities ... Steel ... Retailing ... Automobiles ... Publishing ... OTHERS?
23. Considering your software subcontractors, please circle those industries which represent at least 10% of their US software sales. If there are OTHERS, please specify.

Banking ... Securities ... Steel ... Retailing ... Automobiles ... Publishing ... OTHERS?

24. Considering the packaged software you adapt, please circle those industries which represent at least 10% of their US software sales. If OTHERS, please specify.

Banking ... Securities ... Steel ... Retailing ... Automobiles ... Publishing ... OTHERS?

Accounting ... Engineering ... Data Bases ... Word-processing ... Spread Sheet ... OTHERS?

25. (a) If packaged software costs 10, how much would it cost to adapt to your use?

10  20  30  40  50  More

25. (b) If customized software costs 100, how much would adapted packaged software cost?

10  20  30  40  50  60  70  More

26. Are the managers of the different EDP functions at your major customers’ firm career EDP personnel, or do they come from other parts of the customers’ company for a few years and are then rotated to non-EDP functions? Circle the option which best describes your customers.

Career EDP personnel............Rotated to non-EDP functions.

27. What impact will the US government's role in the Information Highway have on your customers’ EDP Planning? Software Procurement or Development? Other?

28. To what degree will client servers and office computers combined with workstations and PCs replace mainframes in the US market in five years? In ten years?

% of mainframes replaced in five years:

% of mainframes replaced in ten years:

29. (a) In what industries is mainframe or super computer use in the US likely to continue indefinitely?

29. (b) In what functions is mainframe or super computer use in the US likely to continue indefinitely?

30. Do you have any other views comments on the direction or structure of the software industry?

ALL INFORMATION CONTAINED IN THIS QUESTIONNAIRE WILL REMAIN CONFIDENTIAL.

Thank you for your kind cooperation in filling out this questionnaire. Please fax to (212)- 678-6958
Software Industry Questionnaire for User Companies

Name of Company: ________________________ Date: ________________
Address: ________________________________ Tel: _________________
__________________________________ Fax: _________________
__________________________________
Contact: ________________________________

1. Is your firm independent or a subsidiary of a larger company?

2. If a subsidiary, does your parent have other affiliated software users or software producers?

3. How many employees are doing software development?

4. For your firm over each of the last two years what percentage of your total costs in the US were represented by costs related to EDP, such as hardware, personnel, contract employees, software, and EDP consulting services?

5. Out of your US EDP costs, how much was related to software developed by you for your own use?

6. What percentage of the software was purchased from non related companies?

7. How is your EDP operation in the US structured and how many employees do you have working in each EDP function?

8. Does your firm have its own captive US software suppliers?

9. What is their size in terms of sales and staff?

10. What percentage of your software was purchased from such related companies and what percentage from others?

11. What percentage of their sales are to you or your affiliates?

12. What percentage of software bought from others was represented by equipment manufacturers or their affiliates and what percentage by independent software contractors or systems integrators?

13. With how many systems or software operating platforms do you operate? Do they vary by function?

14. With which systems or software operating platforms do you work or interface?

15. Have you ever had to adopt a different platform or system to acquire or use a particular software package?

16. (a) How do you plan for changes in your hardware and software systems for each of your major EDP functions?

16. (b) When you contract outside for software, how is supplier selected?
17. How are system selection decisions made by your subsidiaries or affiliates (e.g. historical usage, current operating platform, latest technology, price reliability)? Do you supply them software functions for mainframe? Networks? PCs/Workstations?
   - Mainframe:
   - Network:
   - PCs/Workstations:

18. Please help us fill in the table below: The two columns represent one, your current mainframe or minicomputer software supplier and two, your other software suppliers; Now for each cell please estimate the percent of your total software purchases respectively for each category:

Category I - Network Servers
Category II - Workstations
Category III - PCs

<table>
<thead>
<tr>
<th></th>
<th>Current Mainframe or Minicomputer software suppliers</th>
<th>Current other software suppliers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Client Servers</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Workstations</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Personal Computers</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

19. Respectively, what percentage of your software expenditures are for super computers, mainframe, workstations, personnel computers, internal telecommunications or networking, and external telecommunications or networking?

<table>
<thead>
<tr>
<th>Percent of your software expenditures?</th>
<th>Super-computers</th>
<th>Mainframe computers</th>
<th>Workstations</th>
<th>personal computers</th>
<th>int. telecom or networking</th>
<th>ext. telecom or networking</th>
</tr>
</thead>
</table>

20. What percentage of application functions are developed by you internally or through software developer affiliates?

21. In each software category, primarily on what do your vendors or vendors' competitors compete?
   - Price
   - Reliability
   - Rapid deployment
   - Technology
   - Interoperability
   - Operating speed
   - Other?

22. (a) Please list major software products you purchase.

22. (b) Please list major software products you develop.

23. For each major product purchased, please describe distribution channel used to acquire software.
   1) Large Integrated Producers
   2) Independent systems integrators
   3) Independent outlets to large users
   4) Independent outlets to small and medium size business users
   5) Retail outlets to individuals

24. Now considering two types of suppliers mentioned in the table below, what percent of your software used in the US in the last two years was developed by:
   1) your company or related subsidiaries
   2) your systems suppliers' own software developers or captive software suppliers?
   3) some other software developers or integrated systems producers?
<table>
<thead>
<tr>
<th>Your company or subsidiary</th>
<th>Current Mainframe or Minicomputer software suppliers</th>
<th>Current other software suppliers</th>
</tr>
</thead>
<tbody>
<tr>
<td>System suppliers’ own or captive software suppliers</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Other developer or integrated systems producers</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

25. Considering your own software development, please circle those industries in which you are an expert? If OTHERS, please specify.

Banking ... Securities ... Steel ... Retailing ... Automobiles ... Publishing ... OTHERS

26. Considering your software subcontractors, please circle those industries in which they are an expert? If OTHERS, please specify.

Banking ... Securities ... Steel ... Retailing ... Automobiles ... Publishing ... OTHERS

27. Please circle those industries and functions in which the supplier of packaged software is an expert If OTHERS, please specify.

Industries: Banking ... Securities ... Steel ... Retailing ... Automobiles ... Publishing ... OTHERS
Functions: Accounting ... Engineering ... Data Bases ... Word-processing ... Spread Sheet ... OTHERS

28. (a) If packaged software costs 10, how much would it cost to adapt to your use?

10  20  30  40  50  More

28. (b) If customized software costs 100, how much would adapted packaged software cost?

10  20  30  40  50  60  70  More

29. What percentage of your total purchases of software for use in the US is:

Packaged
Customized

30. Are the managers of the different EDP functions in your firm career EDP personnel, or do they come from other parts of the your company for a few years and are then rotated to non-EDP functions? Circle the option which best describes your EDP managers.

Career EDP personnel............Rotated to non-EDP functions.

31. What impact will the US government's role in the Information Highway have on your: EDP Planning? Software Procurement or Development? Other?

32. To what degree will client servers and office computers combined with work stations and PCs replace your mainframes in the US in five years? In ten years?

Five years:
Ten years:

33. In what functions will mainframe or super computer use in the US likely continue indefinitely?
34. Do you have any views or comments on the direction or structure of the software industry?

ALL INFORMATION CONTAINED IN THIS QUESTIONNAIRE WILL REMAIN CONFIDENTIAL.

Thank you for your kind cooperation in filling out this questionnaire. PLEASE FAX TO (212) 678-6958
APPENDIX V
EUROPEAN QUESTIONNAIRES
APPENDIX VI

LIST OF COOPERATING FIRMS, GROUPS AND INSTITUTIONS

Aldus (Adobe Systems) Apple Computer Ajinomoto
American Electronics Assoc. ATT Andor Company Ltd.
Boston Consulting Canon Chase Manhattan
Citicorp C. Itoh Columbia University
CSK CREO Inc. Enicom
Fujitsu Fuji-Xerox Fukui Computer
Hewlett-Packard Hitachi Software Engineering Hurwitz Consulting
Information Promotion Agency IBI IBM
IVL Technologies Japan DEC Japan Dataquest
Japan Information Processing Japan Information Service Industry Association Japan Institute of Labor
Development Center Industry Association
Japan PC Software Association Japan Users Association Kanto Electronics Corp.
Marubeni Hytech Corp. Meiji Life Insurance MITI
Mitsubishi Corp Mitsubishi Electric Mitsubishi Real Estate
Mitsubishi Research Institute Mitsui Knowledge Industry Morgan Stanley
Motorola National Center for Science Information Systems National Institute Science
& Technology
Nikko Research Institute Nikko Securities NEC
NTT Data Nissan Motors Nomura Research Institute
Office Naval Research Omron Software Otsuka Shokai
RACE - Tokyo University Rikei Corporation Salomon Brothers
Scudder, Stevens & Clark Sega Sigma Systems
Stanford Kyoto Center Swiss Bank Corporation Symmetrix
Thinking Machines Tokai Bank Toshiba
Toyo Corporation Toyota Trans Cosmos Inc.
US Embassy Tokyo US ITC UnixWare Technology