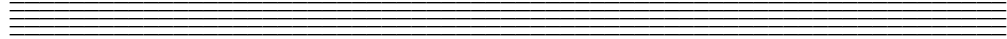


# **A Comparison of U.S. and Japanese Software Process Maturity**



Watts S. Humphrey  
David H. Kitson  
Julia Gale

CMU/SEI-91-TR-027

Approved for public release.  
Distribution unlimited.



# Table of Contents

<b>1. Introduction</b>	<b>1</b>
<b>2. The Japanese Software Industry</b>	<b>3</b>
<b>3. Software Process Background</b>	<b>6</b>
<b>4. SEI Software Process Maturity Model</b>	<b>8</b>
<b>5. Software Process Maturity Questionnaire</b>	<b>11</b>
<b>6. Assessing Software Organizations</b>	<b>12</b>
6.1. SEI-Assisted Assessments	12
6.2. Assessment Tutorials	13
6.3. Other Assessment Methods	13
<b>7. Data Collection and Analysis - U.S. Data</b>	<b>14</b>
7.1. Data Usage Considerations	14
7.2. Data Analysis	15
7.3. Software Process Maturity Level Distribution	15
7.4. Profiles of Negative Responses to Key Questions	18
<b>8. Data Collection and Analysis - Japanese Data</b>	<b>19</b>
8.1. Project Selection	19
8.2. Data Usage Considerations	20
8.3. Data Analysis	21
8.4. Japanese Software Process Maturity Distribution	22
8.5. Profiles of Negative Responses to Key Questions	23
<b>9. Comparative Findings</b>	<b>24</b>
<b>10. Conclusions and Implications</b>	<b>26</b>
<b>11. Acknowledgements</b>	<b>28</b>
<b>References</b>	<b>29</b>
<b>Appendix A: Key Questions and Response Profiles</b>	<b>30</b>

## List of Figures

<b>Figure 1:</b> Software Organization Size Distribution - Japanese Assessment Tutorial Data	3
<b>Figure 2:</b> SEI Software Process Maturity Model	8
<b>Figure 3:</b> Software Process Maturity Level Distribution - U.S. Assessment Tutorial Data	16
<b>Figure 4:</b> Software Process Maturity Level Distribution - U.S. SEI-Assisted Assessment Data	17
<b>Figure 5:</b> Application Area Distribution - Japanese Assessment Tutorial Data	21
<b>Figure 6:</b> Software Process Maturity Level Distribution - Japanese Assessment Tutorial Data	22
<b>Figure 7:</b> Comparison of Software Maturity Distribution - U.S. and Japanese Data	25
<b>Figure A.1:</b> Percent Negative Response of Level 1 Projects to Level 2 Key Questions (Japanese Assessment Tutorial Data - 196 Projects)	38
<b>Figure A.2:</b> Percent Negative Response of Level 1 Projects to Level 2 Key Questions (U.S. Assessment Tutorial Data - 96 Projects)	39
<b>Figure A.3:</b> Percent Negative Response of Level 1 Projects to Level 2 Key Questions (U.S. SEI-Assisted Assessment Data - 41 Projects)	40
<b>Figure A.4:</b> Percent Negative Response of Level 1 Projects to Level 3 Key Questions (Japanese Assessment Tutorial Data - 191 Projects)	41
<b>Figure A.5:</b> Percent Negative Response of Level 1 Projects to Level 3 Key Questions (U.S. Assessment Tutorial Data - 96 Projects)	42
<b>Figure A.6:</b> Percent Negative Response of Level 1 Projects to Level 3 Key Questions (U.S. SEI-Assisted Assessment Data - 41 Projects)	43
<b>Figure A.7:</b> Comparison of Percent Negative Response of Level 1 Projects to Level 2 Key Questions	44
<b>Figure A.8:</b> Comparison of Percent Negative Response of Level 1 Projects to Level 3 Key Questions	45

# List of Tables

<b>Table A.1:</b> Key Questions for Level 2	33
<b>Table A.2:</b> Key Questions for Level 3	34
<b>Table A.3:</b> U.S. and Japanese Level 2 Negative Response Ranking	35
<b>Table A.4:</b> U. S. and Japanese Level 3 Negative Response Ranking	36



# A Comparison of U.S. and Japanese Software Process Maturity

**Abstract:** This report characterizes the software processes currently used by software managers and practitioners in the U.S. and Japan.

The U.S. data for this comparative study of the state of software practice in the U.S. and Japan is drawn from extensive SEI assessments conducted from 1987 through 1989. The Japanese maturity data was gathered during a three-week trip to Japan by the authors. This includes data on 168 U.S. projects and 196 Japanese software projects.

This data is not a statistically valid sample of the software capabilities of either the U.S. or Japanese software industries. However, the data does indicate that the Japanese software industry is, in some respects, both weaker and stronger than that in the U.S. There are also significant differences in the maturity findings between the U.S. and Japanese organizations and these have important implications for software organizations in both countries.

## 1. Introduction

This report represents a comparative study of the state of the software engineering practice in the U.S. and Japan. There has been considerable discussion about the comparative state of U.S. and Japanese software practice with a widespread opinion that the Japanese are ahead. Although there is some justification for this view, this study finds that the Japanese software industry is quite diverse and the state of the practice varies considerably by industry segment. While this study could not comprehensively review this large industry, it does provide process maturity data on some portions and draws on previous studies and reports for some broader conclusions.

The U.S. data in this report is drawn from extensive SEI assessments conducted from 1987 through 1989 as published in [HUM89b]. This includes data on 168 projects together with interviews of nearly a thousand software managers and practitioners. The bulk of these software projects were in industrial organizations working under contract to the U.S. Department of Defense (DoD). A smaller set of projects was drawn from U.S. commercial software organizations and U.S. government software groups. While there is insufficient data to draw definitive conclusions on the relative maturity of these three groups, we have generally found that industrial DoD software contractors have a somewhat stronger software process maturity than either the commercial software industry or the government software groups.

The data on Japanese software was gathered during a 1990 trip to Japan by the authors. During this trip, we visited six Japanese companies and obtained data on 196 software projects in over 88 different software organizations. The Japanese maturity data was generally gathered in advance of our visit. With few exceptions, these organizations are in a software or a software-related business and in many cases, they are working in association with one of the large Japanese computer suppliers to provide software for custom applications.

The Japanese software industry is not monolithic. In fact, one might say that there are two Japanese software industries: one is comprised of a few large, highly competent software factories, while the other has nearly 4,000 small application development groups. Our data indicates that the former (software factories) are equivalent to and possibly stronger than the best U.S. groups, while the latter are below even the lowest level of general U.S. practice. There are also significant differences in the maturity findings between U.S. and Japanese organizations, and these have important implications for software organizations in both countries. These implications are discussed in Section 10 of this report.

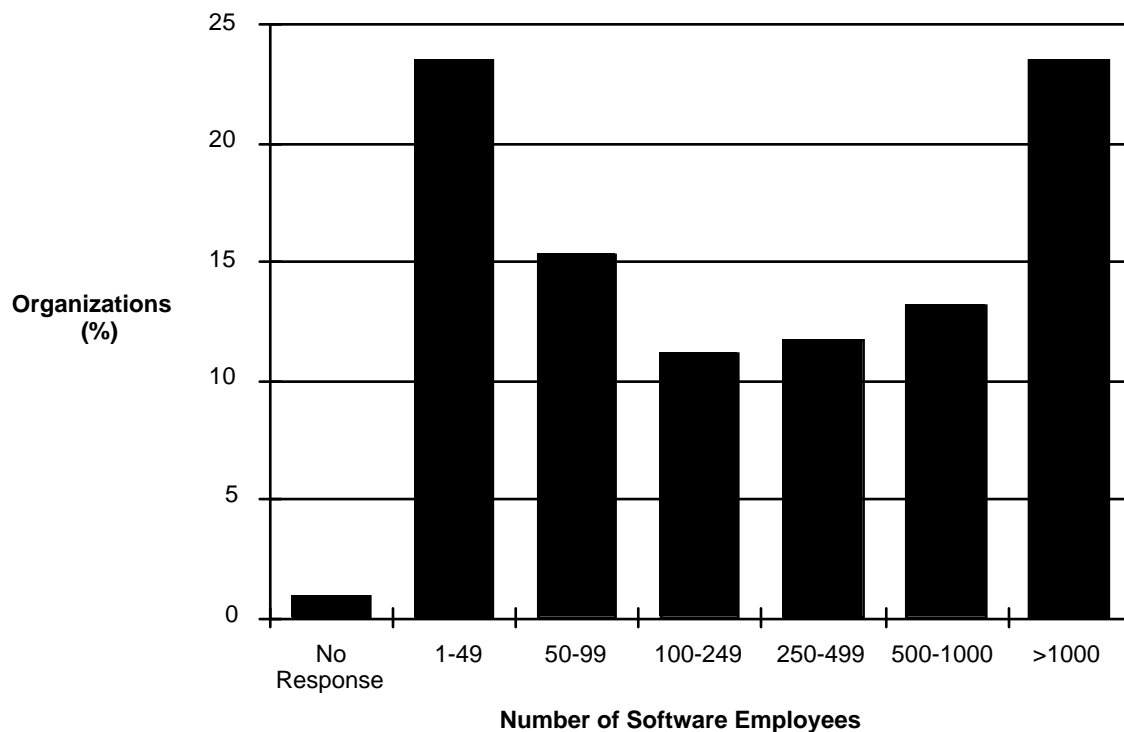
The report characterizes the software processes currently used by software managers and practitioners in the U.S. and Japan. This report is organized into ten sections. Section 2 provides a brief overview of the Japanese software industry. Sections 3 through 6 provide the background and framework for collecting the data upon which the report is based. Sections 7 and 8 describe the data collected and the analyses performed on this data. Sections 9 and 10 discuss implications of these findings for both U.S. and Japanese organizations. Appendix A provides a more detailed analysis of the data collected.



## 2. The Japanese Software Industry

While it is not our intent to provide a comprehensive overview of the large and highly varied Japanese software industry, some background on its structure and practices is necessary to appreciate the findings and conclusion of this study.

Based on published data, there are nearly 4,000 software firms in Japan and 86.7% of them have fewer than 300 total employees [YAM89, pp.4-5]. Figure 1 shows the size (software employees) distribution of the Japanese organizations that have projects that responded to the SEI maturity questionnaire used for this study.



**Figure 1. Software Organization Size Distribution-  
Japan Assessment Tutorial Data (195 Data Points)**

The Japanese software industry has a unique structure that is quite different from that in the U.S. This stems both from the nature of the software business in Japan and from cultural and historical traditions that continue to shape this industry. For various reasons, Japanese firms do not widely use pre-developed software products but rely almost entirely on custom software products. By comparison, 60% of U.S. software and 40% of European software revenue comes from standard products; but in Japan, only 10% is derived from this source. While there is no widely accepted explanation for this distribution, the key reasons appear to be:

- There is no dominant equipment architecture in Japan.
- Since the Japanese place higher value on concrete over abstract items, software is viewed as a "service" that accompanies the hardware.
- Japanese customers are generally reluctant to accept programs that do not precisely fit their needs.
- The established firms in the U.S. and European software industries have given low priority to packaging their products to meet unique Japanese needs.
- The generally high price of packaged software imported to Japan reduces marketability of those products.

Most Japanese software firms are affiliated either with a major computer supplier or with one of the major industrial customers. As a result, the major Japanese computer firms each provide guidance and direction to thousands of software professionals, even though only 10 - 20% of them are on their direct payroll. For example, we interviewed professionals from projects ranging from five to several hundred professionals where fewer than 20% of these workers were on the central corporate payroll. In all cases, however, this small professional core provided both management and technical project direction.

Each of the major computer suppliers has a core software organization, often called a software factory. Such groups are generally staffed by graduates from the leading Japanese universities who work on the most critical software such as operating systems or communication control programs. In each of these companies, a second and generally much larger body of software professionals are devoted to developing customized applications. These professionals form the small central core of projects, which are largely staffed by transient teams of programmers from affiliated subsidiary or subcontract organizations. Often these affiliated software professionals have trade or technical school educations and are managed in a matrix-like structure by marketing-oriented managers.

The reliance of Japan on custom software applications causes or contributes to the following problems:

- The revenue per employee in the Japanese software industry is only about one tenth of the Japanese industry in general [YAM89, p. 5].
- As a low-profit industry, it is also a low-paying one.
- Since Japanese software professionals typically feel they work longer hours than do professionals in other fields, software is viewed as an unattractive occupation.
- Because people are leaving the field and young people are not entering it at the required rate, there is currently a shortage of approximately 600,000 software

professionals [ADA89, p.18]. This shortage is expected to reach 1,000,000 by the year 2000.

- While software is currently the fastest growing Japanese industry, skill shortages are an increasingly severe problem [YAM89, pp. 9-11].

These problems are further exacerbated by the fact that Japanese employees tend to stay with one employer for their entire career. Thus, the only significant source of new software talent is from new university graduates. We also heard from several Japanese sources that software engineering education is generally not available in the leading Japanese universities and that computer science education is quite limited. This combination of factors makes it difficult for Japanese industry to adjust their work force priorities. Since continued growth of Japanese industry requires solutions to these problems, some compromises will be required. In our view, this will likely mean that properly designed and packaged software products will face explosive growth in the Japanese market.

### 3. Software Process Background

The Software Engineering Institute (SEI) was established by the U.S. DoD to focus on improving the state of software development in the U.S. In pursuing this charter, the SEI approach has emphasized the following:

- Developing and validating a software process framework and evaluation methodology for identifying capable organizations.
- Transitioning the evaluation methodology to DoD software acquisition agencies and their prime contractors
- Developing and refining an associated assessment methodology for use by the DoD software community for assessing their software engineering capabilities and determining improvement needs.
- Characterizing and reporting on the state of software engineering practice
- Facilitating software process improvement in U.S. industry

Since early 1987, the SEI Software Process Program has focused on software process as a means of improving the ability of software organizations to produce software products according to plan. This focus on software process is based on the premises that 1) the process of producing and evolving software products can be defined, managed, measured, and progressively improved and 2) the quality of a software product is largely governed by the quality of the process used to create and maintain it.

The software process is the set of activities, methods, and practices that guide people in the production of software. An effective process must consider the relationships of the required tasks, the tools and methods, and the developers' skills, training, and motivation.

Software process management is the application of process engineering concepts, techniques, and practices to explicitly monitor, control, and improve the software process. It is only one of several activities that must be effectively performed for software-producing organizations to be consistently successful. Capable and motivated technical people are critical; knowledge of the ultimate application environment is needed, as is detailed understanding of the end user's needs [CUR88]. Even with all these capabilities, however, inattention to the software management problems will likely result in disappointing organizational performance. A more comprehensive discussion of the role and significance of software process, the discipline of software process management, and software process improvement methods is provided in [HUM89a] and [KIT89].

This view of process and process management has led to the development of a process maturity model, a related software process maturity questionnaire, and a software

process assessment methodology. These form the key elements of SEI's methods for assessing and improving software organizations. Sections 4 through 6 briefly discusses these elements and the methods for applying them to software process improvement.

## 4. SEI Software Process Maturity Model

The software engineering capability of an organization can be characterized with the aid of the software process maturity model shown in Figure 2. This model provides five maturity levels, identifies the key improvements required at each level, and establishes a priority order for moving to higher levels of process maturity.

Level	Characteristic	Key Challenges	Result
<b>5 Optimizing</b>	Improvement fed back into process	Human intensive process Maintain organization at optimizing level	<b>Productivity &amp; Quality</b>     <b>Risk</b>
<b>4 Managed</b>	(Quantitative) Measured process	Changing technology Problem analysis Problem prevention	
<b>3 Defined</b>	(Qualitative) Process defined and institutionalized	Process measurement Process analysis Quantitative quality plans	
<b>2 Repeatable</b>	(Intuitive) Process dependent on individuals	Training Technical practices Process focus	
<b>1 Initial</b>	(Ad hoc / chaotic)	Project management Project planning Configuration management SQA	

Figure 2. SEI Software Process Maturity Model

At the initial level (level 1), an organization can be characterized as having an ad hoc, or possibly chaotic, process. Typically, the organization operates without formalized procedures, cost estimates, and project plans. Even if formal project control procedures exist, there are no management mechanisms to ensure that they are followed. Tools are not well integrated with the process, nor are they uniformly applied. In addition, change control is lax, and senior management is not exposed to or does not understand the key software problems and issues. *When projects do succeed, it is generally because of the heroic efforts of a dedicated team rather than the capability of the organization.*

An organization at the repeatable level (level 2) has established basic project controls: project management, management oversight, product assurance, and change control. The strength of the organization stems from its experience at doing similar work, but it faces major risks when presented with new challenges. The organization has frequent quality problems and lacks an orderly framework for improvement.

At the defined level (level 3), the organization has laid the foundation for examining the process and deciding how to improve it. A Software Engineering Process Group (SEPG) has been established to focus and lead the process improvement efforts, to keep management informed on the status of these efforts, and to facilitate the introduction of a family of software engineering methods and technologies.

The managed level (level 4) builds on the foundation established at the defined level. When the process is defined, it can be examined and improved but there is little data to indicate effectiveness. Thus, to advance to the managed level, an organization establishes a minimum set of measurements for the quality and productivity parameters of each key task. The organization also establishes a process database with resources to manage and maintain it, to analyze the data, and to advise project members on its meaning and use.

Two requirements are fundamental to advance from the managed to the optimizing level (level 5). Data gathering should be automated, and management should redirect its focus from the product to process analysis and improvement. At the optimizing level, the organization has the means to identify the weakest process elements and strengthen them, data are available to justify applying technology to various critical tasks, and numerical evidence is available on the effectiveness with which the process has been applied. The key additional activity at the optimizing level is rigorous defect cause analysis and defect prevention.

These maturity levels have been selected because they do the following.

- Reasonably represent the historical phases of evolutionary improvement of actual software organizations.
- Represent a measure of improvement that is reasonable to achieve from the prior level.
- Suggest interim improvement goals and progress measures.
- Make obvious a set of immediate improvement priorities, once an organization's status in this framework is known.

While there are many aspects to the transition from one maturity level to another, the basic objective is to achieve a controlled and measured process as the foundation for continuous improvement.

It has been our experience (based on ten SEI-assisted assessments conducted from 1987 through 1989) that when software organizations are assessed against this maturity framework, the assessment method enables reasonably accurate placement of them on the maturity scale and helps to identify key improvement needs. In practice we find that when management focuses on the few highest priority items, their organizations generally make rapid improvement in being able to produce quality software products on time and within budget. While the use of tools and technology can enhance software

engineering capability, such investments are generally of limited value for organizations with low-maturity software processes.

[HUM89a] and [KIT89] provide more comprehensive descriptions of software process management and the maturity model.



## 5. Software Process Maturity Questionnaire

The maturity questionnaire is a structured set of yes-no questions that helps to facilitate the conduct of reasonably objective and consistent assessments of software organizations [HUM87]. It has also been designed to assist DoD acquisition organizations in identifying software contractors with acceptable software engineering capabilities. Since the instrument and method for applying it are publicly available, software contractors can use them to identify areas for improvement.

The questions in the maturity questionnaire cover three areas.

1. Organization and resource management. This area deals with functional responsibilities, personnel, and other resources and facilities.
2. Software engineering process and its management. This area is concerned with the scope, depth, and completeness of the software engineering process and the way in which the process is measured, managed, and improved.
3. Tools and technology. This area deals with the tools and technologies used in the software engineering process. It helps determine the effectiveness with which the organization employs basic tools and methodologies.

Some sample questions from the maturity questionnaire are:

- Is there a software engineering process group or function?
- Is a formal procedure used to make estimates of software size?
- Are code and test errors projected and compared to actuals?

## 6. Assessing Software Organizations

There are a number of ways the software process framework (software process concepts and principles, the maturity model, and the maturity questionnaire) can be applied. The SEI has developed, and has experience with, the following.

- SEI-assisted assessments
- assessment tutorials
- self-assessments
- SEI-licensed vendor assessments
- capability evaluations

While all these methods have contributed to the SEI's data on the software process, to remain consistent with our published U.S. data, the data in this report was obtained from SEI-assisted assessments and assessment tutorials. The paragraphs below briefly discuss each of these. A more comprehensive discussion of how assessments are conducted and the role of assessment in improving software engineering capability is contained in [KIT89].

### 6.1. SEI-Assisted Assessments

An SEI-assisted assessment is an appraisal of an organization's current software process by a trained team of experienced software professionals. Typically, a team is composed of four to six SEI professionals and one to three professionals from the organization being assessed. A methodology for conducting assessments has been developed by the SEI [OLS89]. The assessment team receives training in the methodology prior to conducting the actual assessment. The goal is to facilitate improvement of the organization's software process. The assessment team identifies the most important software process issues currently facing the organization and develops recommendations to deal with them. Since the objective is improvement, validation of questionnaire responses (e.g., requesting substantiating documents) is limited to those topics that directly affect progress to the next higher level of process maturity.

SEI-assisted assessments are conducted in accordance with an assessment agreement signed by the SEI and the organization being assessed. This agreement provides for senior management involvement, organizational representation on the assessment team, confidentiality of results, and follow-up actions.

The SEI has conducted such assessments since February 1987 and is using the information gained to refine its knowledge of the state of the practice of software engineering in the DoD software community. [HUM89b] is the first state of the software engineering practice report published based on the framework discussed in Sections 4 through 6.

## 6.2. Assessments Tutorials

At assessment tutorials, professionals from various organizations learn about process management concepts, assessment techniques, and the SEI assessment methodology. They also supply demographic data on themselves and their organizations as well as on a project for which they complete an assessment questionnaire. This format is designed for people who wish to learn more about the SEI assessment methodology with minimal investment.

The data collected at assessment tutorials is added to the SEI assessment database and is used for various analyses. Assessment tutorials are typically conducted at conferences and symposia attended by software professionals from the DoD and DoD contractors (e.g., National Security Industrial Association, Electronic Industries Association, and the annual SEI affiliates symposium).

## 6.3. Other Assessment Methods

Self-assessments are similar to SEI-assisted assessments, with the primary difference being assessment team composition. Self-assessment teams are composed primarily of software professionals from the organization being assessed, with possibly one or two SEI software professionals present. The context, objective, and degree of validation are the same as for SEI-assisted assessments.

Vendor-assisted assessment are SEI assessments that are conducted under the guidance of commercial vendors who have been trained and licensed to do so by the SEI. The assessment team is trained by the vendor and consists of software professionals from the organization being assessed plus at least one vendor professional who has been qualified by the SEI. (Licensing commercial vendors to offer these assessment services provides a means of making SEI software process assessments available to a wider audience than the SEI would otherwise be capable of doing.)

Capability evaluations, like SEI-assisted assessments and self-assessments, are appraisals of an organization's current software process; however, the context, purpose, and assessment team composition are different. The context of capability evaluation is the DoD acquisition process, and the purpose is to provide information concerning the organization's software engineering capabilities for the acquisition agency. This information is then considered, along with other relevant information, in the source selection decision. Hence, validation of maturity questionnaire responses is a greater consideration here than it is in assessments.

## 7. Data Collection and Analysis - U.S. Data

The U.S. material for this report is based on information of two types:

- Responses to the maturity questionnaire [HUM87] and a questionnaire designed to collect demographic data. The responses were collected from assessment tutorials and SEI-assisted assessments.
- The collective knowledge and experience that the SEI has acquired as a result of our involvement in the development and application of the various assessment methods discussed in Section 6.

Assessment participants include software and hardware developers from DoD organizations, DoD contractors, and commercial enterprises. The data in this report is taken from ten organizations that participated in SEI-assisted assessments (with four to six projects involved in each assessment), and over 70 organizations were represented in the assessment tutorials, representing 168 data points<sup>1</sup> from across the U.S. [HUM89b]. In every assessment, the SEI signs an agreement that there will be no attribution of the results to a specific company. The implications and recommendations presented in this report, therefore, represent an aggregate view.

### 7.1. Data Usage Considerations

The U.S. results described reflect the state of the software engineering practice based on the data, experience, and knowledge as described in [HUM89b]. There are, however, some important considerations relating to the data gathering and analytical approach.

First, the sample was not statistically selected. Most of the respondents came from organizations that are affiliated with the SEI. These respondents varied in the type and degree of involvement with the projects they reported.

Another consideration is the degree of validation of the responses; the extent to which corroboration of responses was requested depended on the type of assessment being conducted. While we have not precisely measured the effect of this factor, the distributions seem roughly similar and the issue priorities of each level are comparable. Thus we judge the consequences of this factor to be modest.

In comparing the questionnaire responses received from assessment tutorials and from SEI-assisted assessments, several points should be noted.

---

<sup>1</sup>A *data point* is one set of yes-no responses to the maturity questionnaire; the scope of these responses is a specific software project.

- The SEI-assisted assessments were conducted on-site by a trained team, with participation from knowledgeable project managers and technical professionals.
- The assessment tutorial respondents contained a mixture of management and non-management professionals, some of whom likely had detailed knowledge of the technical points.
- For SEI-assisted assessments, many threshold responses were verified.
- For assessment tutorials, the respondents were able to ask questions, however, no responses were verified.

## 7.2. Data Analysis

Two views of the data were prepared and analyzed: (1) software process maturity level distribution and (2) percent negative response to key questions<sup>2</sup>. For the purposes of this report, we separate the response data from the SEI-assisted assessments and the assessment tutorials, and treat them as two distinct data populations. Because of the previously noted statistical considerations, we do not believe that a greater depth of analysis than that which is presented in this report is justified.

## 7.3. Software Process Maturity Level Distribution

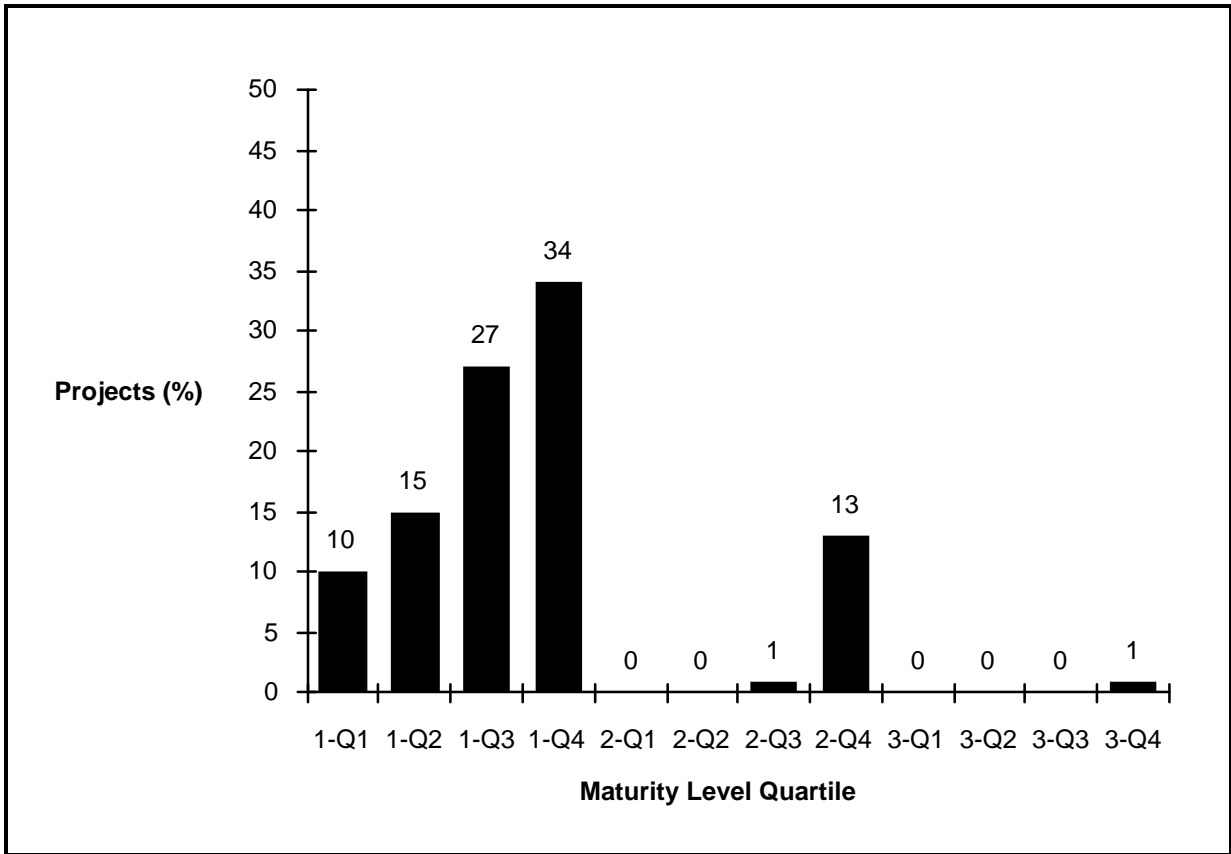
The distribution of software process maturity level across the sample provides a high-level view of the state of the practice. Figure 3 shows the software process maturity distributions for assessment tutorials and Figure 4 shows the data for SEI-assisted assessments.

For both figures, the vertical axis represents the percentage of data points in the population; the horizontal axis represents the software process maturity scale—levels 1 through 5. To show additional structure, the maturity scale has been further divided into quartiles—four quartiles for each maturity level for a total of 20 quartiles<sup>3</sup>. Because of the limited data at levels 4 and 5, only the lowest 12 quartiles are shown. The quartiles are identified in the charts using the notation  $x - Q_y$ , where  $x$  is the level (1-5), and  $y$  is the quartile (1-4). In Figure 3, for example, 2 - Q4 refers to the fourth (and last) quartile for level 2 and contains approximately 13% of the sample. It should be noted that these quartiles are used solely for the purpose of providing low-maturity organizations with a more precise way to evaluate their improvement progress. Thus a level 1 organization in any quartile would be of level 1 and would still perform that way.

---

<sup>2</sup>Key questions are those for which a high percent of affirmative responses is required to qualify for a particular maturity level.

<sup>3</sup>Each data point's location in the maturity level distribution was determined by the number of additional affirmative responses needed to rate the project at the next higher process maturity level. The range of these values was then divided into four "buckets" or quartiles.



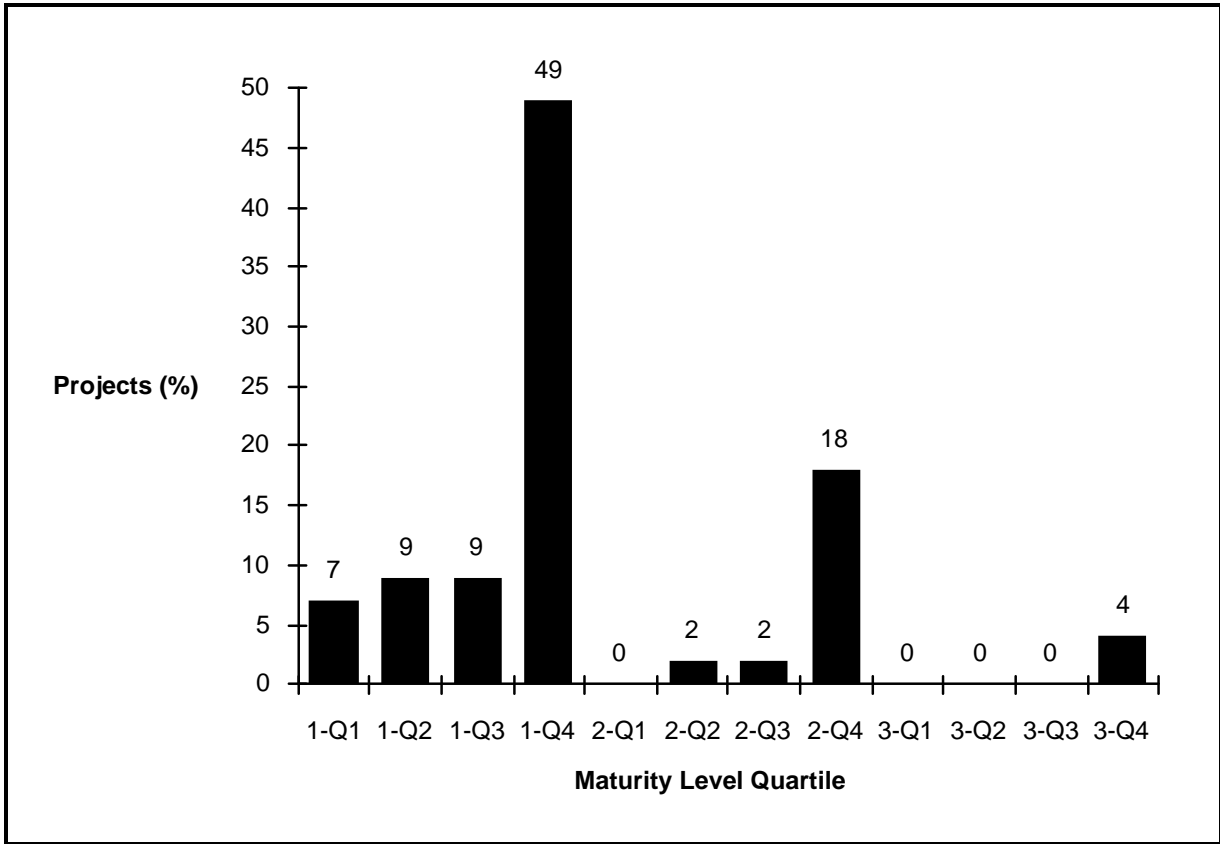
**Figure 3. Software Process Maturity Level Distribution- U.S. Assessment Tutorial Data (113 Data Points)<sup>4</sup>**

The assessment tutorial results shown in Figure 3 indicate that the majority of the respondents reported projects at the initial level of maturity. Figure 3 shows a large percentage of the respondents to be in the fourth quartile of level 1 (quartile 1 - Q4); with minimal improvement, these projects could be classified as level 2. Fourteen percent of all the tutorial respondents reported projects at the repeatable level (level 2), and only 1% of those respondents described projects at level 3, the defined level. No tutorial respondents reported projects at either the managed level (level 4) or the optimizing level (level 5) of software process maturity.

The maturity level distribution for projects reviewed by SEI-assisted assessments, shown in Figure 4, is very similar to that of the tutorial data. Although tutorial participants were largely mid- to upper-level managers not currently managing a project (as opposed to the project managers who provided data for SEI-assisted assessments), the profiles of process maturity are surprisingly similar. Some key differences are apparent, however. First, Figure 4 shows that the SEI-assisted population is skewed slightly towards higher levels of process maturity. Secondly, a large numbers of projects

<sup>4</sup>Note that the percentages may not total 100 due to rounding errors.

in quartile 4 of maturity levels 1, 2, and 3 are poised for moving to the next higher maturity level.



**Figure 4. Software Process Maturity Level Distribution  
U.S. SEI-Assisted Assessment Data (55 Data Points)<sup>5</sup>**

<sup>5</sup>Note that the percentages may not total 100 due to rounding errors.

## 7.4 Profiles of Negative Responses to Key Questions

The key questions are those that are considered the most important indicator for a particular maturity level. Each question is associated with a particular maturity level; for example, the question "Is a formal procedure used to make estimates of software size?" is a level 2 question. This means that an organization that has all the attributes of a level 2 software organization (with respect to the SEI process maturity model) would respond affirmatively to this question.

To analyze the responses to key questions, we determined the population percentage responding negatively to each key question for levels 2 and 3 and displayed the results in decreasing order in accordance with the Japanese data. These computations were performed for both the assessment tutorial data and for the SEI-assisted data. The results are provided in Appendix A.



## 8. Data Collection and Analysis - Japanese Data

Because of the size and complexity of the U.S. and Japanese software industries, any reasonably competent comparison must consider data and judgements from many sources. We have drawn heavily from published sources, personal contact, our observations in the U.S., and from seven trips to Japan by the authors.

The data gathering process in Japan followed a somewhat different process to that used in the U.S. While six organizations were visited and examined in some detail, these were one or two day visits and did not follow the disciplined process of a full assessment. In the bulk of the cases, the data gathered was from large meetings where there was insufficient time to interview the individuals in detail. In these cases, the data gathering method was similar to that described for tutorials in the previous sections. A smaller proportion of the Japanese data (7%) was gathered in private meetings where some interaction was possible. Due to the limited time available and the language constraints, these discussions were not as comprehensive as a typical SEI-assisted assessment in the U.S. As a result, we do not have the detailed understanding of the Japanese software process that would result from personal interviews with several hundred practitioners. It is thus possible that these results could be biased in ways we cannot detect. It is our view, however, that the Japanese respondents attempted to represent their process fairly and objectively.

We did note one important bias in the Japanese questionnaire responses. In the U.S., when respondents do not understand a question or are unsure of the answer, they typically leave the answer blank or fill in the "no" entry. In Japan, because of some confusion in the questionnaire instructions, they invariably filled in "yes." Unless we had some independent means to verify these responses, such as thorough interviews, we did not change the Japanese responses. This means that the Japanese maturity profile may be slightly biased, showing a higher maturity level than is actually the case.

### Project Selection

The U.S. data was gathered from a broad sample of industry, with a strong bias toward DoD suppliers and computer manufacturers. The business application community is thus seriously underrepresented. In Japan, we were limited to those organizations to which we were introduced that were willing to provide us data. This sample included many business application programming groups, a number of communications and military suppliers, and only two computer manufacturers (Company A and Company B). Of these two manufacturers, only Company A was willing to provide us with questionnaire responses. Our conclusions regarding these large firms are thus based on the data gathered from Company A, discussions with software people from both organizations, previous trips to Japan, and a large amount of published data on this segment of the Japanese software industry.

## 8.2. Data Usage Considerations

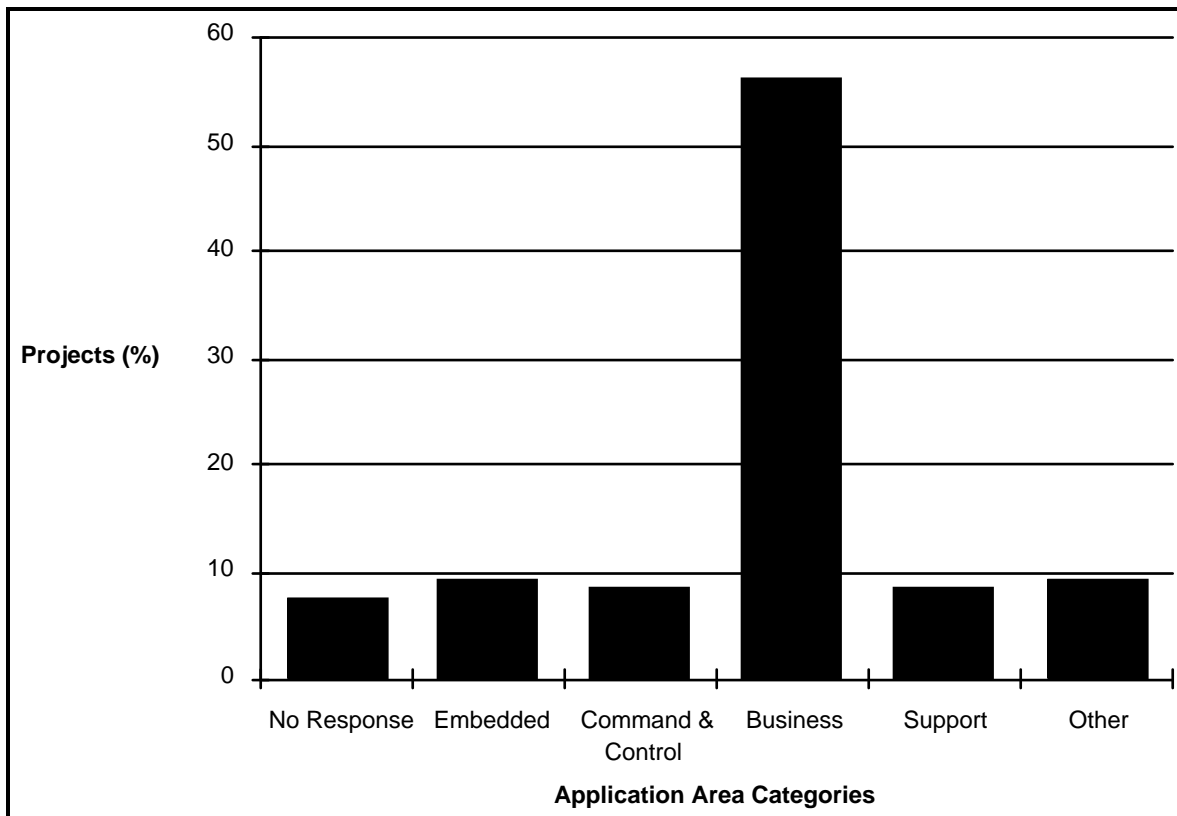
Several cautions should be observed in interpreting this Japanese software process maturity data. First, while a large volume of data was gathered in Japan, it was obtained through the auspices of a relatively few organizations who obtained it for us from their own employees, their customers, and other associated organizations. Regardless of organizational affiliation, all respondents were software professionals. Other than the points noted above, we are not aware of any statistical bias in the data. While it is extremely difficult to evaluate the statistical accuracy of such data, it is our opinion that the information and data we have obtained provides a reasonable picture of the state of the software practice in Japan. It also is likely that several major Japanese software organizations could have substantially stronger processes than indicated by this study. This point is discussed further in the Conclusions and Implication's section.

There are also cultural and language differences that could have caused difficulty in understanding the Japanese responses to our questions. We did not, however, find this to be a serious problem. Most of our hosts were fluent in English and they were already quite familiar with the SEI maturity model and other relevant U.S. work. The reverse concern, however, is more significant. That is, the questions were originally written in English and translated by one of the Japanese organizations with which we dealt. In several cases, we subsequently learned that this translation did not faithfully portray the questions' intent. We also found several areas where our terminology was not familiar to the Japanese. This was particularly true regarding job titles and organizational structure. While this could introduce a bias to our results, we do not believe this had a serious effect.

### 8.3 Japanese Data Analysis

Figure 5 shows the distribution of application areas for the Japanese software projects surveyed by the SEI. In Japan, there is a relatively small military systems business and the embedded and command and control categories generally refer to industrial process control systems rather than weapons systems. As can be seen, business applications constitute the largest single segment of these responses.

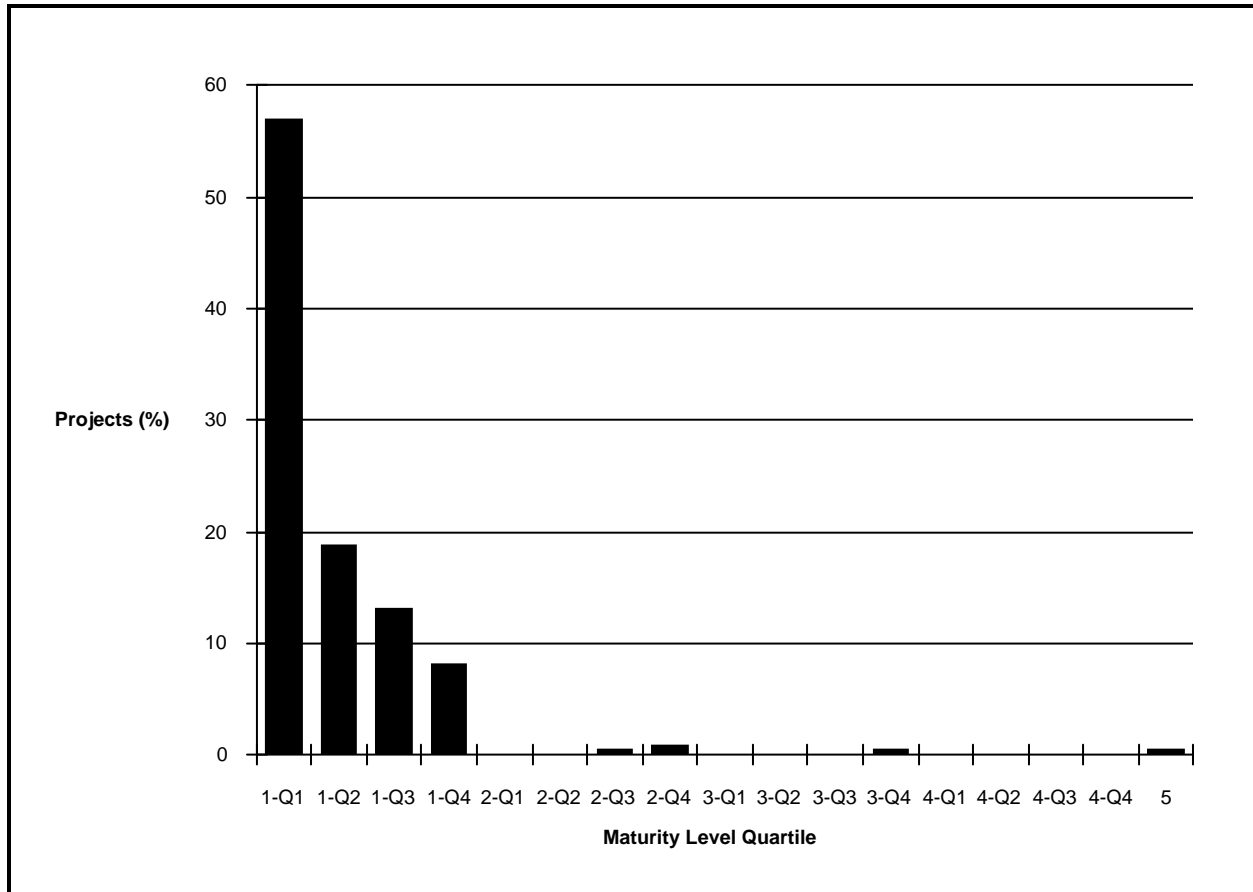
In the following paragraphs, the Japanese response data is analyzed by maturity level. Because of the limited data available, no detailed analysis is warranted of any of the maturity levels above 1.



**Figure 5. Application Area Distribution-  
Japanese Assessment Tutorial Data (195 Data Points)**

## 8.4 Japanese Software Process Maturity Distribution

The maturity distribution for the Japanese responses is shown in Figure 6.



**Figure 6. Software Process Maturity Level Distribution-  
Japanese Assessment Tutorial Data (196 Data Points)**

As can be seen, almost all responses are at level 1. We did, however, obtain one response at level 5 and one at level 2. The level 5 response was for an operating system development group in Company A (the only such group studied). We held several hours of discussions with this level 5 group and concluded that their practices are clearly at level 5 and are equivalent to anything we have seen in the U.S.

As noted previously, the other major computer manufacturer, Company B did not provide us with any questionnaire responses nor the opportunity for detailed project interviews. We did, however, spend a morning in presentations and discussions with eight software people from seven different groups within this company. Of the 21 questions and comments raised by this group, seven inferred a knowledge characteristic of the questioner's process maturity level. Five of these were at level 1, one at level 2, and one at level 3 or above. This last questioner was from their software factory works while

all the other attendees were from application groups, headquarters, or technology departments. It is thus our opinion that the software factory in this company has a maturity capability similar to that observed in Company A.

In addition, Cusumano has gathered substantial data on the software factories in these leading Japanese firms [CUS89, p.29]. He finds that they have an impressive history of process improvements that rivals anything in the U.S. For example he reports that Hitachi doubled productivity in one year, reduced late projects from 72% to 12%, and reduced bugs per machine in the field by eight times. Toshiba better than doubled productivity and reduced bugs by up to seven times. NEC improved productivity by 26% to 91% and reduced bugs by one third. Fujitsu improved productivity by two thirds and reduced bugs by 19 times in eight years.

While we did not have the opportunity for detailed discussions with software factory groups in the other computer firms, we feel it is likely that their practices are similarly advanced. We cannot say, however, whether any other such groups are also at level 5.

With the exception of the core operating systems groups, all the other software groups we interviewed were at level 1 with one at level 2. This was true even for those groups that worked with or for the major computer suppliers.

## **8.5 Profiles of Negative Responses to Key Questions**

To analyze the responses to key questions, we determined the population percentage responding negatively to each key question for levels 2 and 3 and displayed the results in decreasing order. The results are provided in Appendix A. As noted above, there is insufficient Japanese data on projects above level 1 to warrant analyses of their responses.

## 9. Comparative Findings

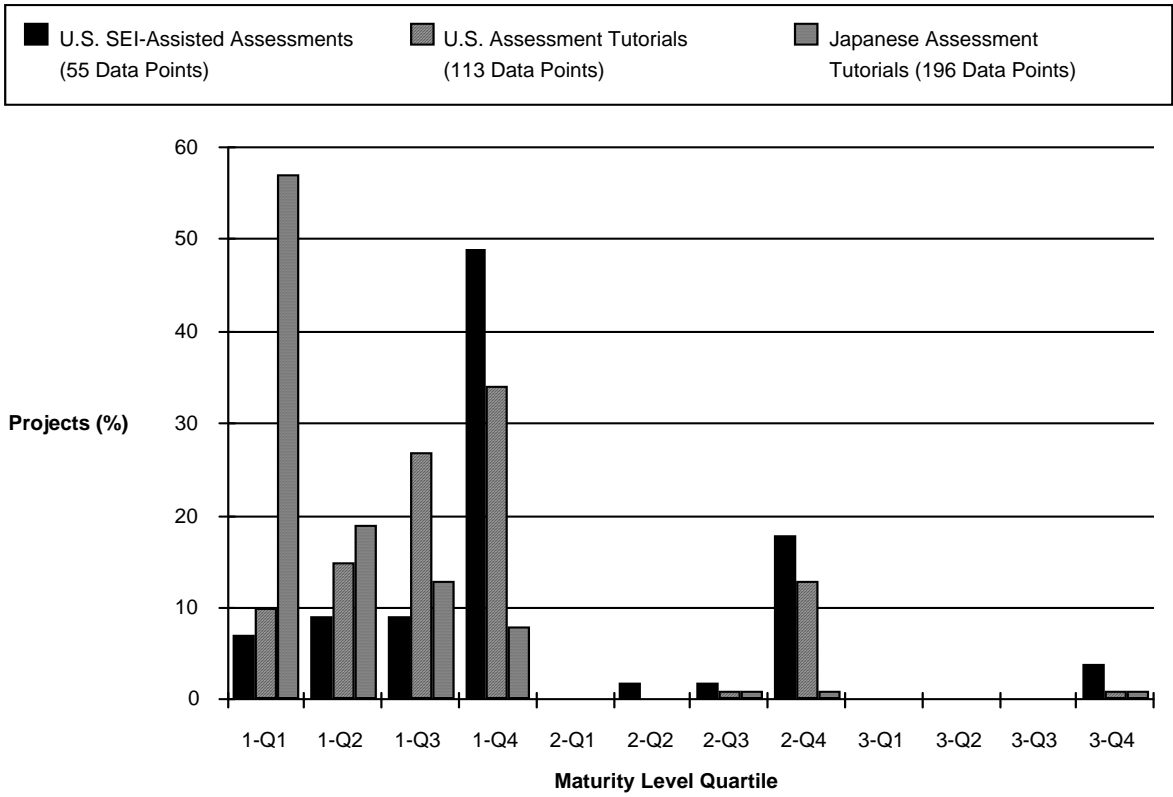
In comparing U.S. and Japanese data, we did note one bias that could be responsible for much of the reported difference between the U.S. and Japanese software maturity levels. In the U.S., most of the SEI maturity data has been obtained from DoD contractors. While the largest portion of these respondents are clustered near the top of level 1, some are at level 2 and above. The limited SEI data on those U.S. commercial groups who develop business applications show these projects all cluster at the bottom of level 1 with no projects at level 2. The bulk of the Japanese software projects surveyed are in this category. While we do not have sufficient data on U.S. business application programming groups to make a definitive comparison, we believe that this factor would partially explain the lower maturity level of the Japanese groups.

A comparison of the U.S. and Japanese maturity level distributions is shown in Figure 7 (which only shows data up to level 3). Only one level 5 project has been assessed in Japan and two in the U.S.<sup>6</sup> With a couple of exceptions, the remainder of the Japanese projects are at level 1. In the U.S., there is a small but growing population of projects at levels 2 and 3. It should be noted that in the U.S., the bulk of the projects are clustered near the top of level 1 and should thus be able to advance to level 2 with modest effort.

In Japan, the highest proportion of the projects are clustered at the bottom of level 1. Such organizations typically have many project management, quality, and configuration management practices to adopt before they can reach level 2. As is typical of organizations at this maturity level, the Japanese software community commonly experiences significant schedule delays and cost overruns.

---

<sup>6</sup>The level 5 project assessed in the U.S. were subsequent to [HUM89b]. These projects are mentioned here because they are relevant to the conclusions drawn in this report.



**Figure 7. Comparison of Software Process Maturity Distribution U.S. and Japanese Data**

## 10. Conclusions and Implications

The U.S. and Japanese software industries are quite different in both structure and products. Based on our findings only a few general conclusions apply to both.

- With only minor differences in terminology, the software maturity model and the SEI questionnaire appear to apply equally well to software groups in both countries.
- With limited exceptions, we have found the level of software practice in both countries is very low. This is not only a problem for the individual organizations involved, but it is also a matter of growing national importance: the continued poor management of software resources will seriously retard the application of potentially beneficial computer technologies in both countries.
- Some organizations in both countries have achieved very high levels of process maturity. This demonstrates that the maturity improvements are effective in both countries and that they both have the technical and management capability to work at this level.

For the Japanese software industry, a few additional conclusions can be drawn.

- The suspected high capability of a few of the core software groups (software factories) in the leading computer firms is a national strength. Based on our limited observations, it appears that this strength is not being transitioned to the rest of the software industry.
- The large branch of the Japanese software industry that develops custom software appears to be both poorly managed and supported. Without focused process and quality management resources this entire industry will likely remain at level 1 for the indefinite future.
- Because of the limited software engineering and computer science curricula of Japanese universities, they are unlikely to solve the growing manpower shortages with new graduates.
- The current weak state of Japanese software practice is of growing national concern. The Ministry of International Trade and Industry (MITI) has recently declared computer software to be of the highest national importance [WAT90].
- Widespread use of prepackaged software is probably the only effective near-term answer to Japan's software resource and process maturity problems. This conclusion appears to be generally accepted in Japan and it is expected that this will be the fastest growing segment in this rapidly growing Japanese industry [ADA89, p.18].

Some conclusions from this study for the U.S. software industry are as follows.

- The view that the Japanese software industry is ahead of the U.S. is unfounded.



- In the critical area of computer systems software, the small elite programming groups in the leading Japanese computer manufacturers appear to be on a par with and possibly even ahead of the best U.S. practice.
- In the critical area of packaged software products, the U.S. has a clear worldwide lead but these firms are ignoring the Japanese market [TOT90]. If history is any guide, the Japanese will establish a strong domestic market and use it as a springboard to achieve world-wide dominance.

The U.S. propensity to blame the Japanese for its trade problems is both misplaced and destructive [TOT90]. U.S. products in Japan are generally clearly identifiable: the controls and instructions are in English and the products are designed to fit U.S. customs and conveniences. Conversely, the most successful Japanese products in the U.S. are generally indistinguishable from those of U.S. manufacturers. The best example is the automobile. Even though the Japanese drive on the left, U.S. automobiles in Japan are equipped as in the U.S. Japanese automobiles in the U.S., however, match our customs and practices. While the U.S. currently enjoys a commanding lead in packaged software, patched up products or partially translated manuals will not satisfy the Japanese market. To address this potentially important area, U.S. suppliers must make the necessary investments to understand Japanese needs and meet them.

The current U.S. industrial position in software is dependent upon the weakness of our competitors. Because of MITI's recent position on the importance of computer software, increased attention should be expected in Japan. As they have in the past, Japan is now adopting the new U.S. technology of software process improvement. If history is any guide, Japan will adopt it more rapidly than the U.S. and our industrial position will be exposed.

## 11. Acknowledgements

In gathering the material for this report, the authors are deeply indebted to many individuals and groups in Japan who provided us with extensive information on their industry and their software practices. We are particularly grateful for the gracious support and hospitality of Mr. Tim Aihara of DEC Japan, and Kouichi Kishida and Edgar Cooke of SRA. We also appreciate the help of Steve Masters and Elynor Helman with the data analysis; the support provided by the SEI librarians Karola Fuchs, Sheila Rosenthal, and their assistants; as well as the invaluable assistance of Dorothy Josephson with the manuscript.

## References

- [ADA89] "Japanese Software, The Next Competitive Challenge," prepared by Dewey, Ballantine, Bushby, Palmer, and Wood for ADAPSO's International Marketing Seminar, January 1989.
- [CUR88] Curtis, B., Krasner, H., Iscoe, N., "A Field Study of the Software Design Process for Large Systems," *Communications of the ACM*, 31 (11), November 1988, pp. 1268-1287.
- [CUS89] Cusumano, M., "The Software Factory: A Historical Interpretation," *IEEE Software*, March 1989, pp. 23-30.
- [HUM89a] Humphrey, W.S., *Managing the Software Process*, Addison-Wesley, Reading, MA, 1989.
- [HUM89b] Humphrey, W. S., Kitson, D. H., and Kasse, T. C., *The State of Software Engineering Practice: A Preliminary Report*, Software Engineering Institute, (CMU/SEI-89-TR-1), February 1989.
- [HUM87] Humphrey, W.S., Sweet, W., et al., *A Method for Assessing the Software Engineering Capability of Contractors*, Software Engineering Institute, (CMU/SEI-87-TR-23, ADA187230.), September 1987.
- [KIT89] Kitson, D.H., Humphrey, W.S., *The Role of Assessment in Software Process Improvement*, Software Engineering Institute, (CMU/SEI-89-TR-3), December 1989.
- [OLS89] Olson, T.G., Humphrey, W.S., Kitson, D.H., *Conducting SEI-Assisted Software Process Assessments*, Software Engineering Institute, (CMU/SEI-89-TR-7), February 1989.
- [TOT90] Totten, B., "Red Paper - Are We Abandoning Our Computer Industry to Japan,?" ComputerServe, 1990.
- [TOT89] Totten, B., "Japan's Mythical Trade Surplus," *New York Times*, Dec. 9, 1990, Section F, p. 13. Interview.
- [WAT90] Watanabe, T., "Japan's Trying Hard to Catch the U.S. in Software," *Los Angeles Times*, July 8, 1990.
- [YAM89] Yamamoto, K., "Japan's Software Industry," *Japanese Computer Quarterly*, Japan Information Processing Development Center (JIPDEC), Tokyo, No. 76, 1989.

# Appendix A. Key Questions and Response Profiles

This section of the report provides a view of selected portions of the response data from assessment tutorials, SEI-assisted assessments, and Japanese responses. Figure A.1 shows the negative response profile for the Japanese to the level 2 key questions. Figures A.2 and A.3 show negative response profiles for the U.S. to the level 2 key questions. Table A.1 provides the text of the same key questions indexed by control number. For example, Figure A.2 shows that of the assessment tutorial projects reported to be at level 1 (96 out of a total of 113 projects), 64% responded negatively to question C46 ("Are profiles of software size maintained for each software configuration item, over time?"). This is a key question for advancing to level 2.

Figures A.4, A.5, and A.6 show negative response profiles for the level 3 key questions. Table A.2 provides the text of the same key questions indexed by control number.

Figure A.7 shows both the U.S. and Japanese level 2 response data for the level 1 projects. As can be seen, the Japanese projects have substantially lower (poorer) scores than the U.S. organizations. As noted previously, however, we suspect that the scores would be much more comparable between these two countries if the responses were grouped by application area.

There are also some interesting differences between the U.S. and the Japanese data. For example, the five lowest scores for the U.S. SEI-assisted assessment organizations, Figure A.3, were the following questions (in descending order):

- 42 - Is a formal procedure used to make estimates of software size?
- 44 - Are formal procedures applied to estimating software development cost?
- 43 - Is a formal procedure used to produce software development schedules?
- 24 - Is a formal procedure used in the management review of each software development prior to making contractual commitments?
- 48 - Are statistics on software code and test errors gathered?

For the Japanese organizations, the five questions with the lowest scores were:

- 42 - Is a formal procedure used to make estimates of software size?
- 6 - Does the Software Quality Assurance (SQA) function have a management reporting channel separate from the software development project management?
- 46 - Are profiles of software size maintained for each software configuration item, over time?

14 - Is there a software configuration control function for each project that involves software development?

43 - Is a formal procedure used to produce software development schedules?

While both groups reported a general lack of size estimation methods and procedures for developing schedules, otherwise their major problem areas differed. By ranking the questions in order of percent negative responses for the U.S. and Japanese organization as shown in Table A.3, one can see that the two questions where the Japanese organizations had relatively greater difficulty than the U.S. organizations were:

6 - Does the Software Quality Assurance (SQA) function have a management reporting channel separate from the software development project management?

14 - Is there a software configuration control function for each project that involves software development?

These topics all relate to broader organizational capabilities where the Japanese tended to have less stable and formal arrangements. From our interviews it appears that the widespread use of subcontractors and matrix-like project management structures tends to make SQA and SCM more difficult to implement effectively.

The two questions where the U.S. organizations had relatively more difficulty than the Japanese groups were:

44 - Are formal procedures applied to estimating software development cost?

84 - Do software development first-line managers sign-off on their schedules and cost estimates?

Here the differences are likely more cultural. In the U.S., low maturity software groups are likely to plow ahead without bothering to develop schedules and plans. In Japan, the ingrained practice of consensus management generally requires agreement from all members of the involved groups. This frequently involves some discussions of schedules and estimates.

For level 3 responses in Figure A.8, a similar analysis gives the negative response rankings in Table A.4. Here, interestingly, four of the five worst response questions were common between the two countries:

98 - Is a mechanism used for verifying that the samples examined by Software Quality Assurance are truly representative of the work performed?

20 - Is a formal training program required for design and code review leaders?

99 - Is there a mechanism for assuring the adequacy of regression testing?

15 - Is there a software engineering process group?

These questions are difficult for many software groups, regardless of nationality. Beyond these four common questions, the level 3 area where the Japanese groups seemed to have relatively greater difficulty than the U.S. groups was:

59 - Are the action items resulting from design review tracked to closure?

The question where the US. group had relatively greater difficulty was:

19 -. Is there a required software engineering training program for software developers?

While we can see no simple explanation for these differences, we suspect that the greater U.S. use of configuration control systems and the generally greater Japanese emphasis on training are likely responsible.

## Table A.1: Key Questions for Level 2

CN <sup>7</sup>	Question
6	Does the Software Quality Assurance (SQA) function have a management reporting channel separate from the software development project management?
14	Is there a software configuration control function for each project that involves software development?
24	Is a formal procedure used in the management review of each software development prior to making contractual commitments?
42	Is a formal procedure used to make estimates of software size?
43	Is a formal procedure used to produce software development schedules?
44	Are formal procedures applied to estimating software development cost?
46	Are profiles of software size maintained for each software configuration item over time?
48	Are statistics on software code and test errors gathered?
77	Does senior management have a mechanism for the regular review of the status of software development projects?
84	Do software development first-line managers sign off on their schedules and cost estimates?
87	Is a mechanism used for controlling changes to the software requirements?
96	Is a mechanism used for controlling changes to the code? (Who can make changes and under which circumstances?)

---

<sup>7</sup>CN = *control number* - the control number uniquely identifies a particular question and is invariant across versions of the assessment instrument. These questions are presented here in control number order.

## Table A.2: Key Questions for Level 3

CN	Question
15	Is there a software engineering process group function?
19	Is there a required software engineering training program for software developers?
20	Is a formal training program required for design and code review leaders?
23	Does the software organization use a standardized and documented software development process on each project?
47	Are statistics on software design errors gathered?
59	Are the action items resulting from design reviews tracked to closure?
61	Are the action items resulting from code reviews tracked to closure?
83	Is a mechanism used for ensuring compliance with the software engineering standards?
91	Are internal software design reviews conducted?
92	Is a mechanism used for controlling changes to the software design?
95	Are software code reviews conducted?
98	Is a mechanism used for verifying that the samples examined by Software Quality Assurance are truly representative of the work performed?
99	Is there a mechanism for assuring the adequacy of regression testing?

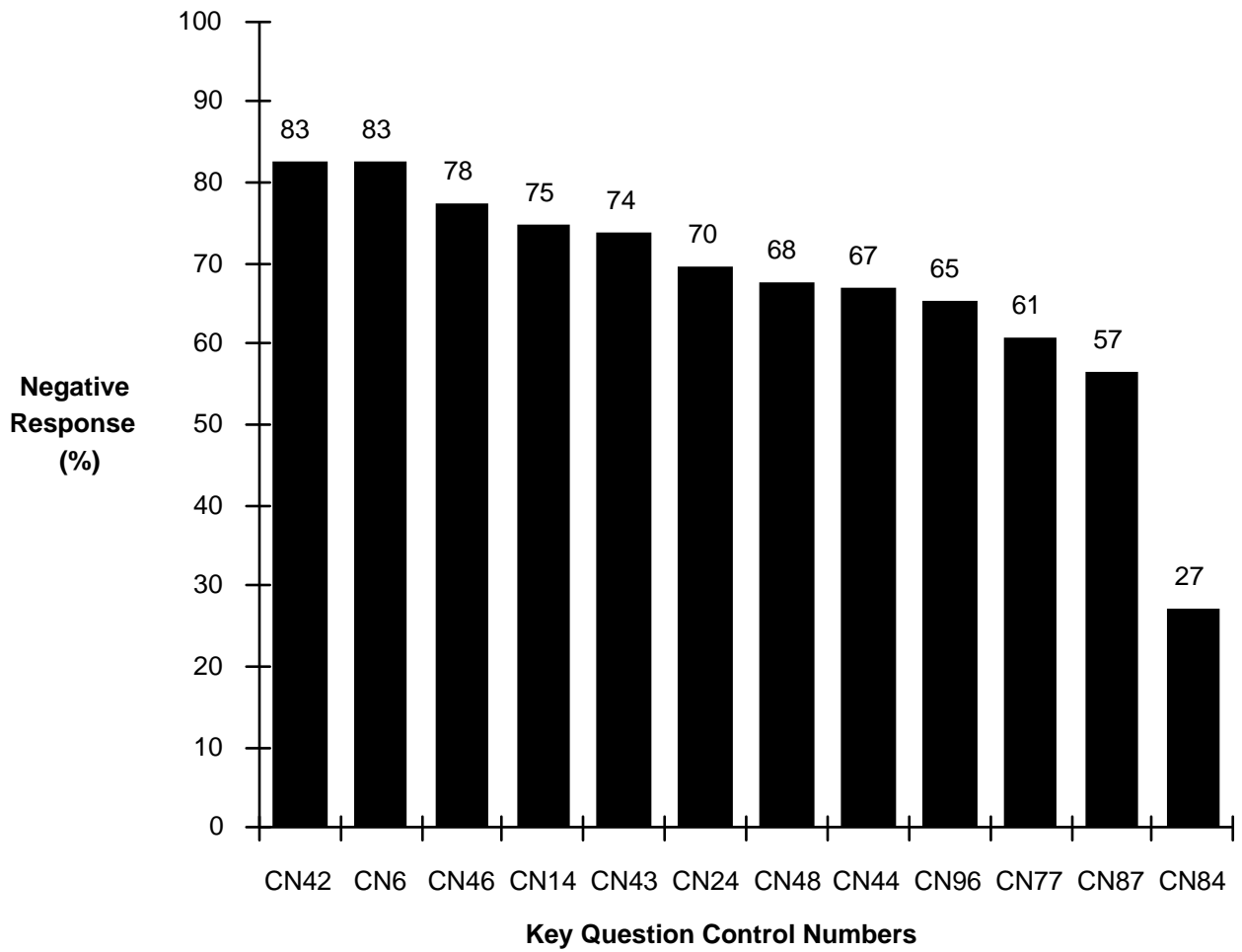


**Table A.3: U.S. and Japanese Level 2 Negative Response Ranking**

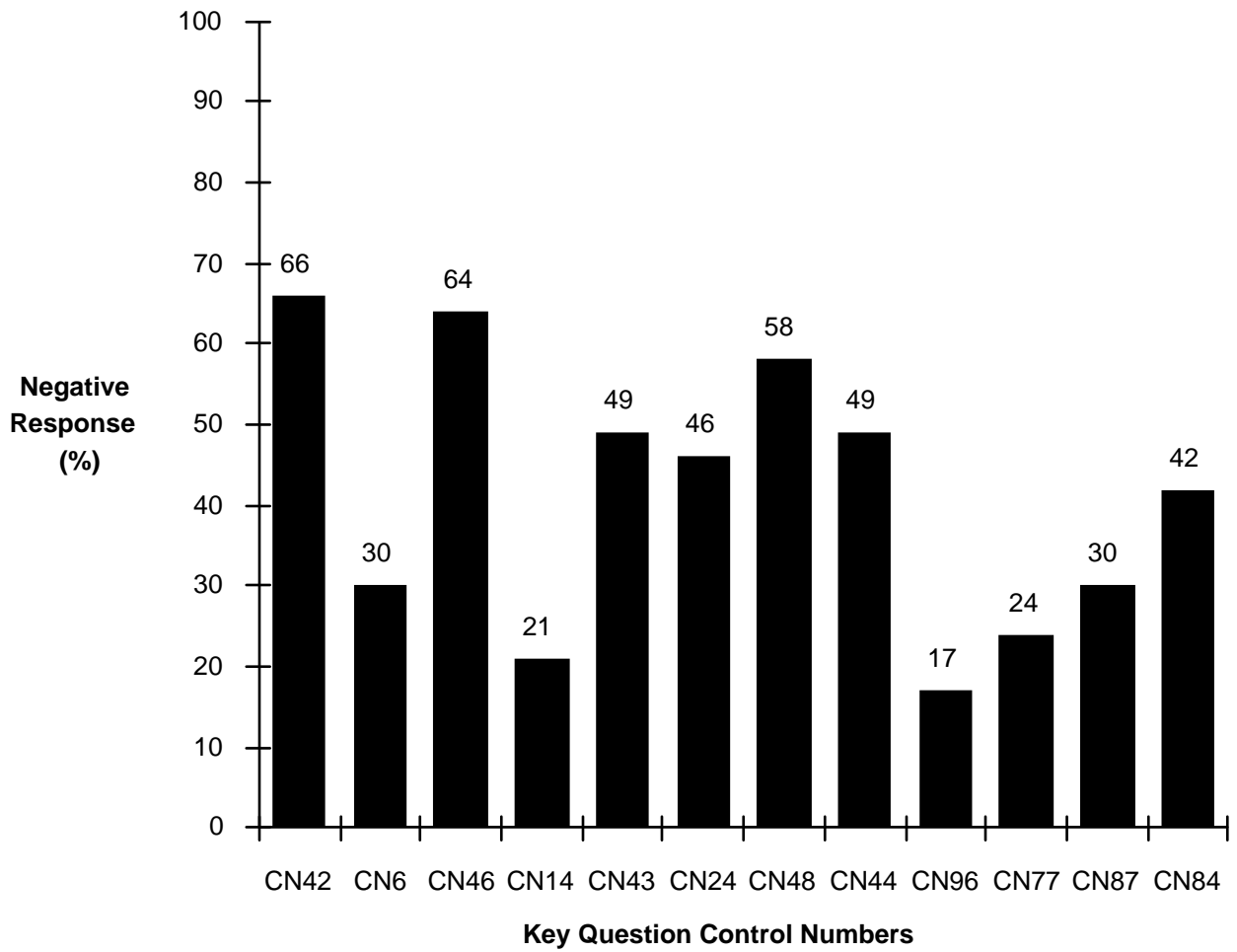
CN	Japan	U.S. Assess- ment Tutorial	U.S.SEI Assisted Assess- ment	Difference Japan and U.S. Assessment Tutorial	Difference Japan and SEI Assisted Assessment
42	1 (=2)	1	1	0	0
6	2 (=1)	8 (=9)	9	-6	-7
46	3	2	6	1	-3
14	4	11	12	-7	-8
43	5	5 (=4)	3	0	2
24	6	6	4	0	2
48	7	3	5	4	2
44	8	4 (=5)	2	4	6
96	9	12	11	-3	-2
77	10	10	8	0	2
87	11	9 (=8)	10	2	1
84	12	7	7	5	5

**Table A.4: U.S. and Japanese Level 3 Negative Response Ranking**

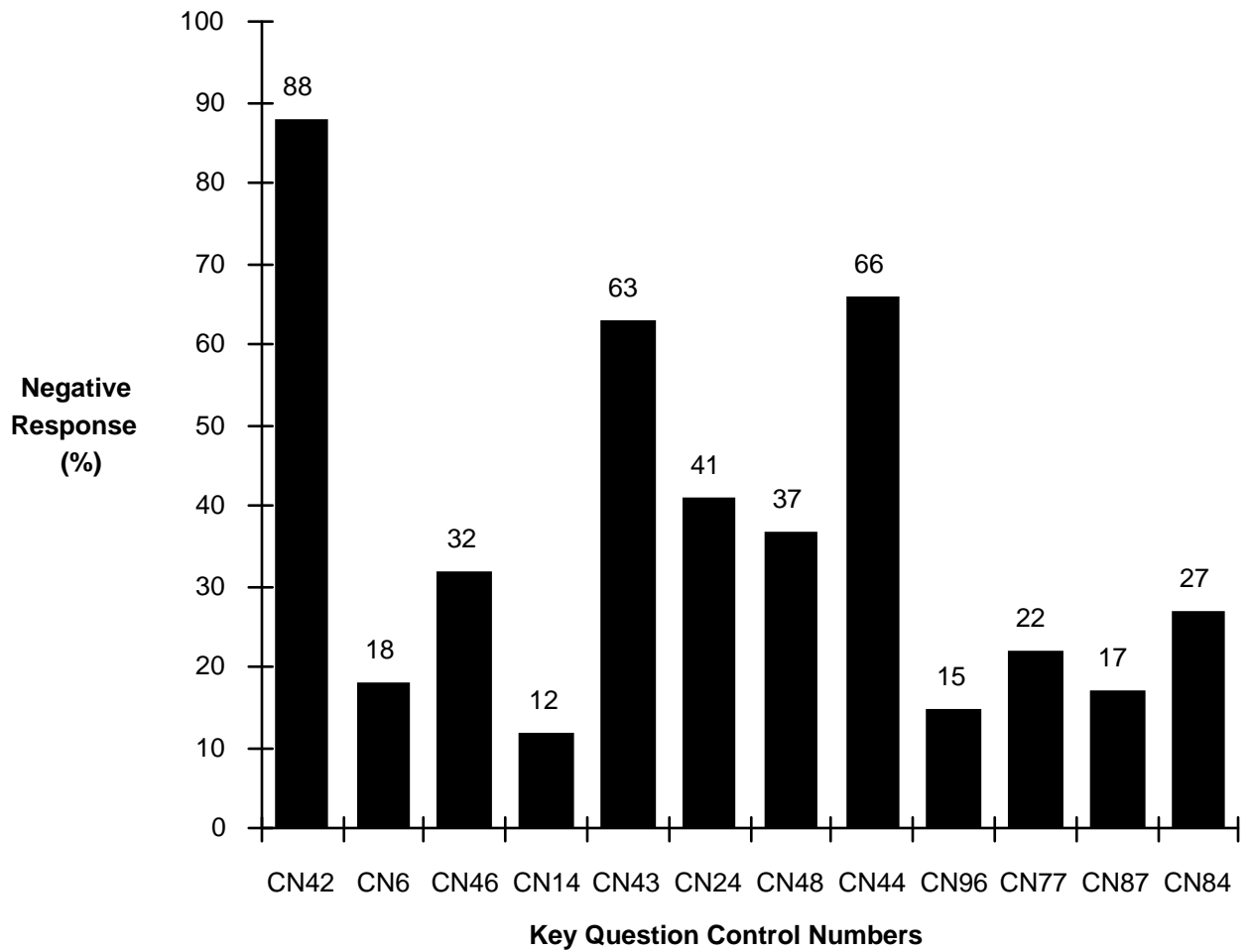
CN	Japan	U.S. Assess- ment Tutorial	U.S.SEI Assisted Assess- ment	Difference Japan and U.S. Assessment Tutorial	Difference Japan and SEI Assisted Assessment
98	1	3	5	-2	-4
20	2	1	1	1	1
99	3	4	4	-1	-1
15	4	5	3	-1	1
83	5	7	7 (=8)	-2	-2
47	6	6	6	0	0
61	7	8 (=9)	9	-1	-2
59	8	12	12 (=13)	-4	-4
19	9	2	2	7	7
95	10	11	8 (=7)	-1	2
92	11	10	10	1	1
23	12	9 (=8)	11	3	1
91	13	13	13 (=12)	0	0



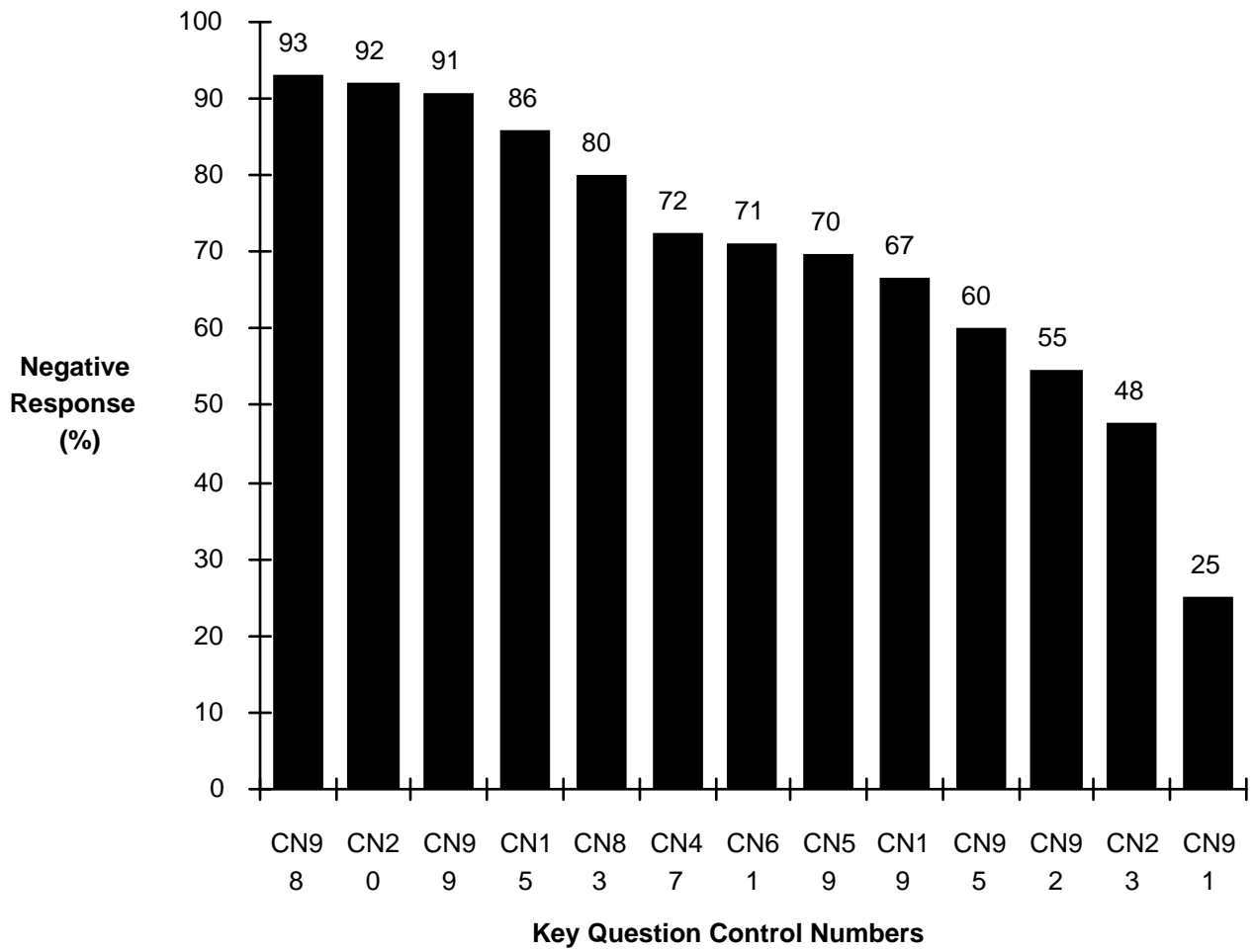
**Figure A.1. Percent Negative Response of Level 1 Projects to Level 2 Key Questions  
(Japanese Assessment Tutorial Data - 196 Projects)**



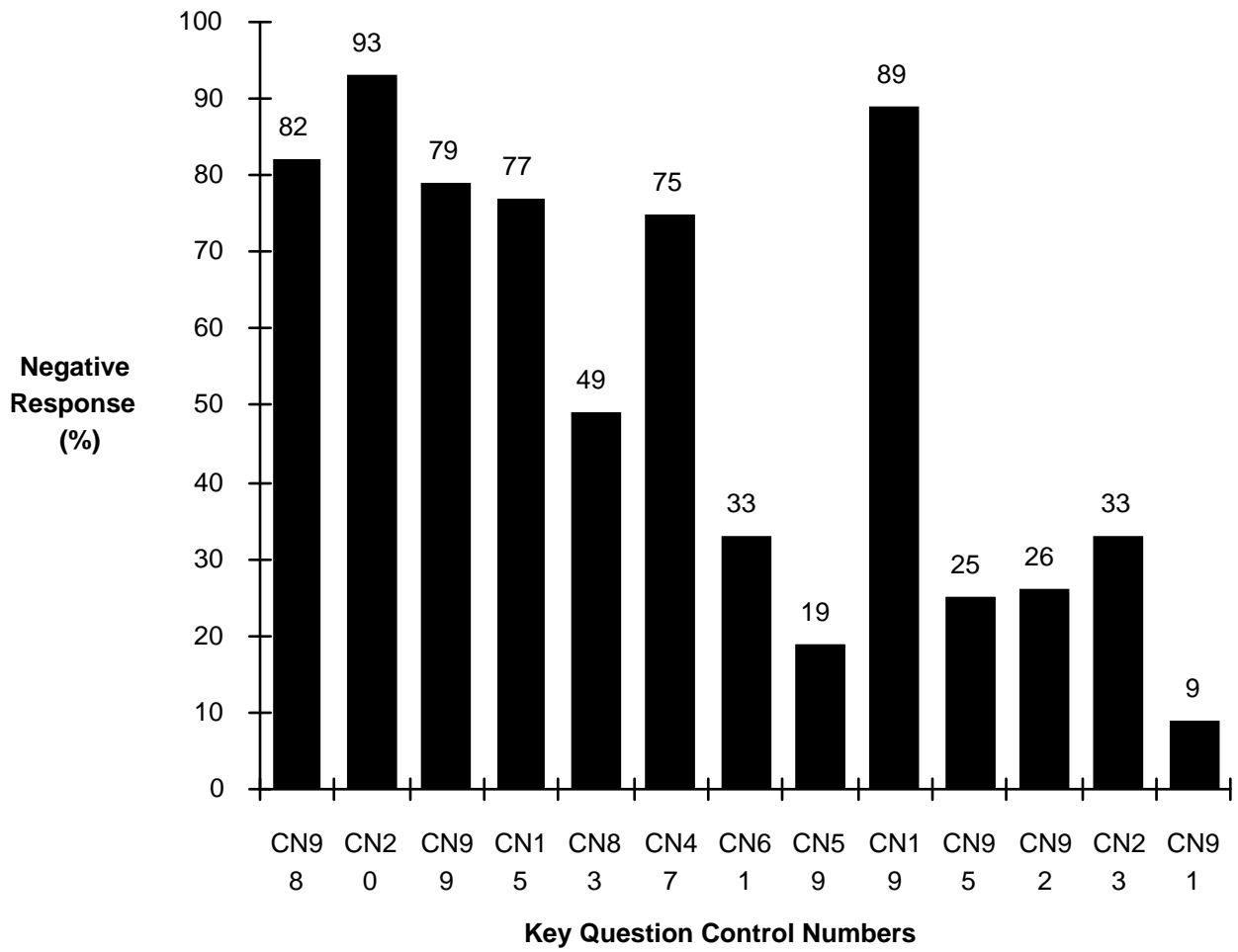
**Figure A.2. Percent Negative Response of Level 1 Projects to Level 2 Key Questions (U.S. Assessment Tutorial Data - 96 Projects)**



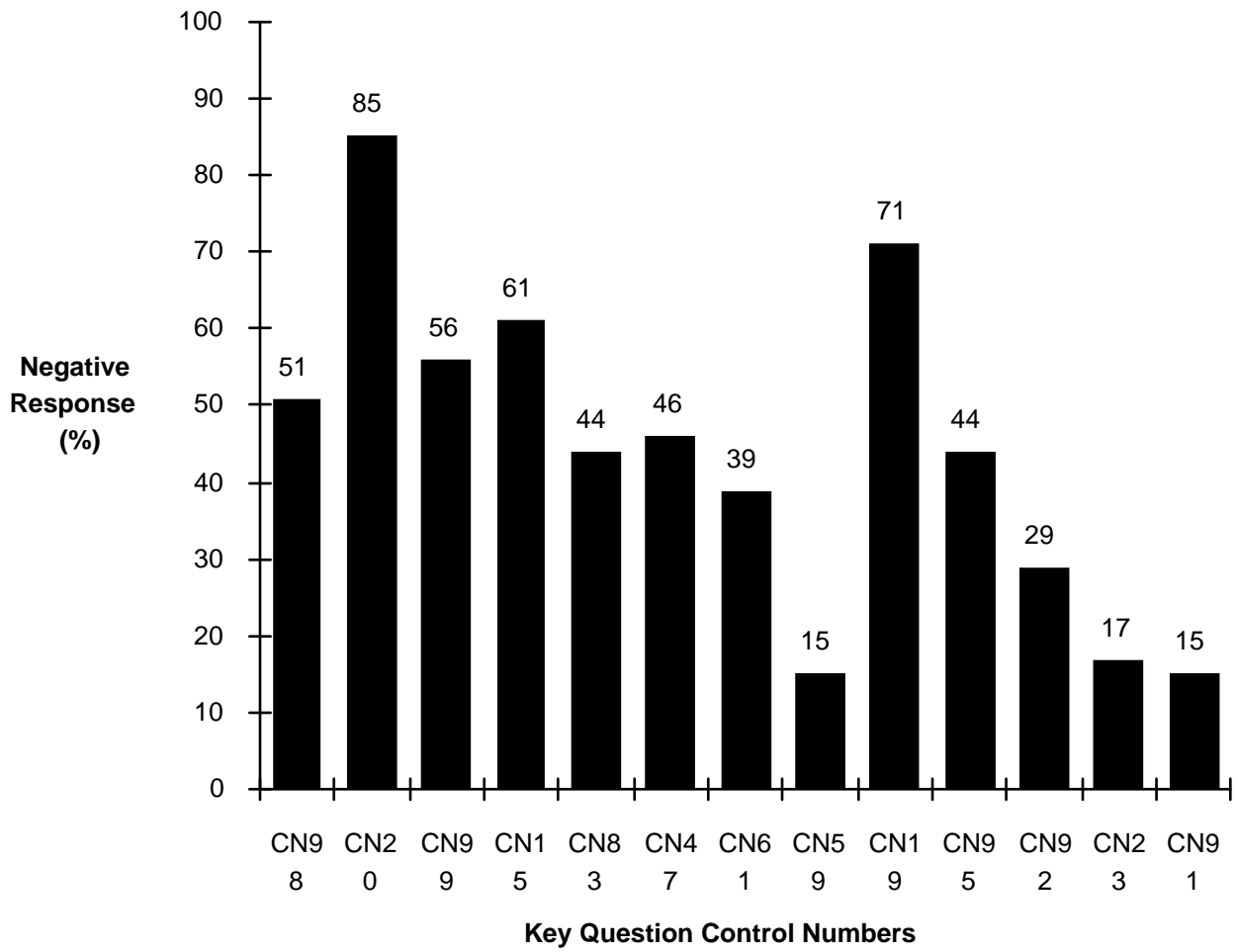
**Figure A.3. Percent Negative Response of Level 1 Projects to Level 2 Key Questions (U.S. SEI-Assisted Assessment Data - 41 Projects)**



**Figure A.4. Percent Negative Response of Level 1 Projects to Level 3 Key Questions  
(Japanese Assessment Tutorial Data - 191 Projects)**

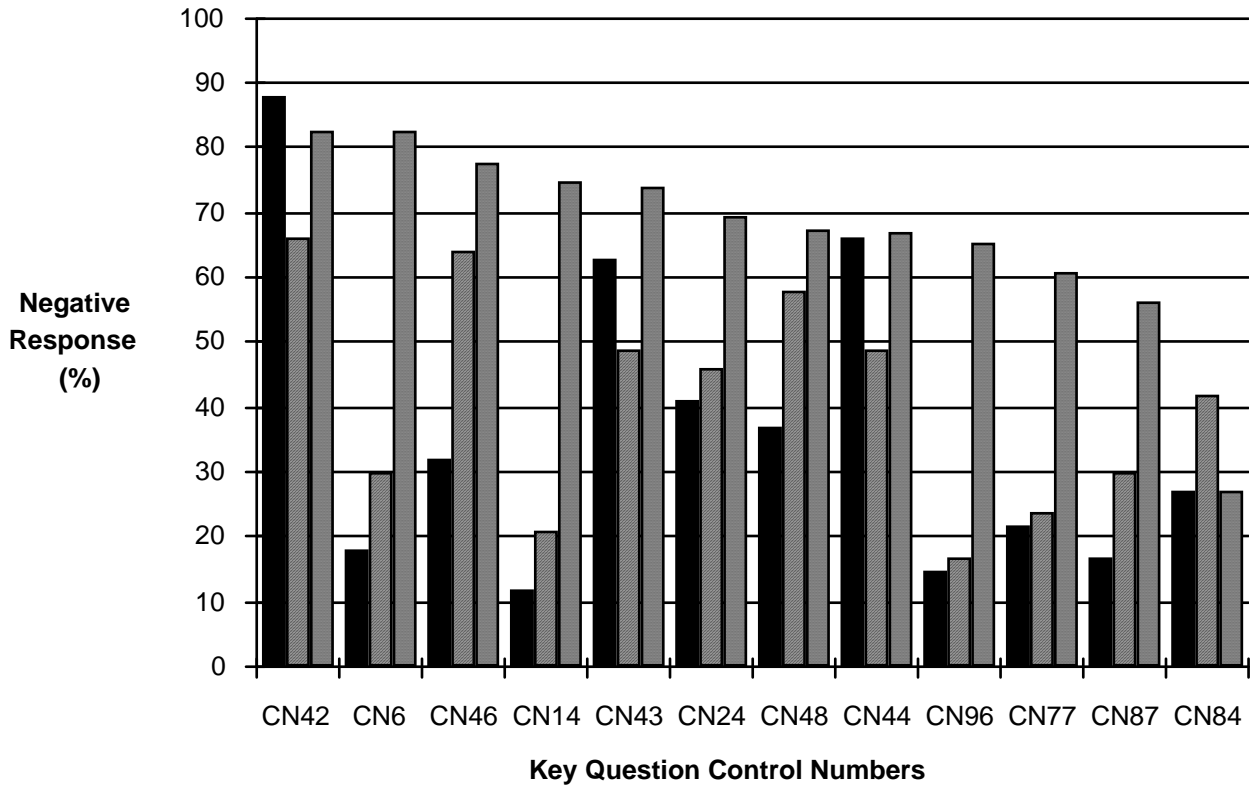
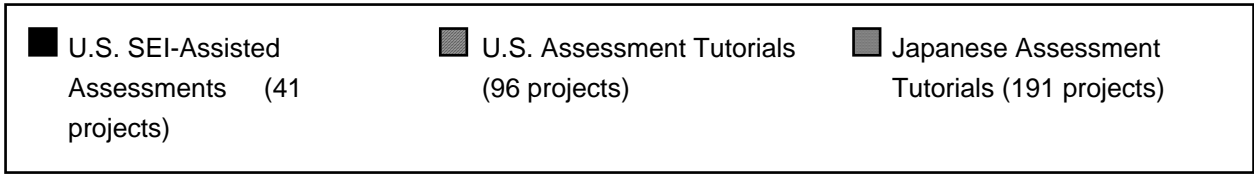


**Figure A.5. Percent Negative Response of Level 1 Projects to Level 3 Key Questions (U.S. Assessment Tutorial Data - 96 Projects)**

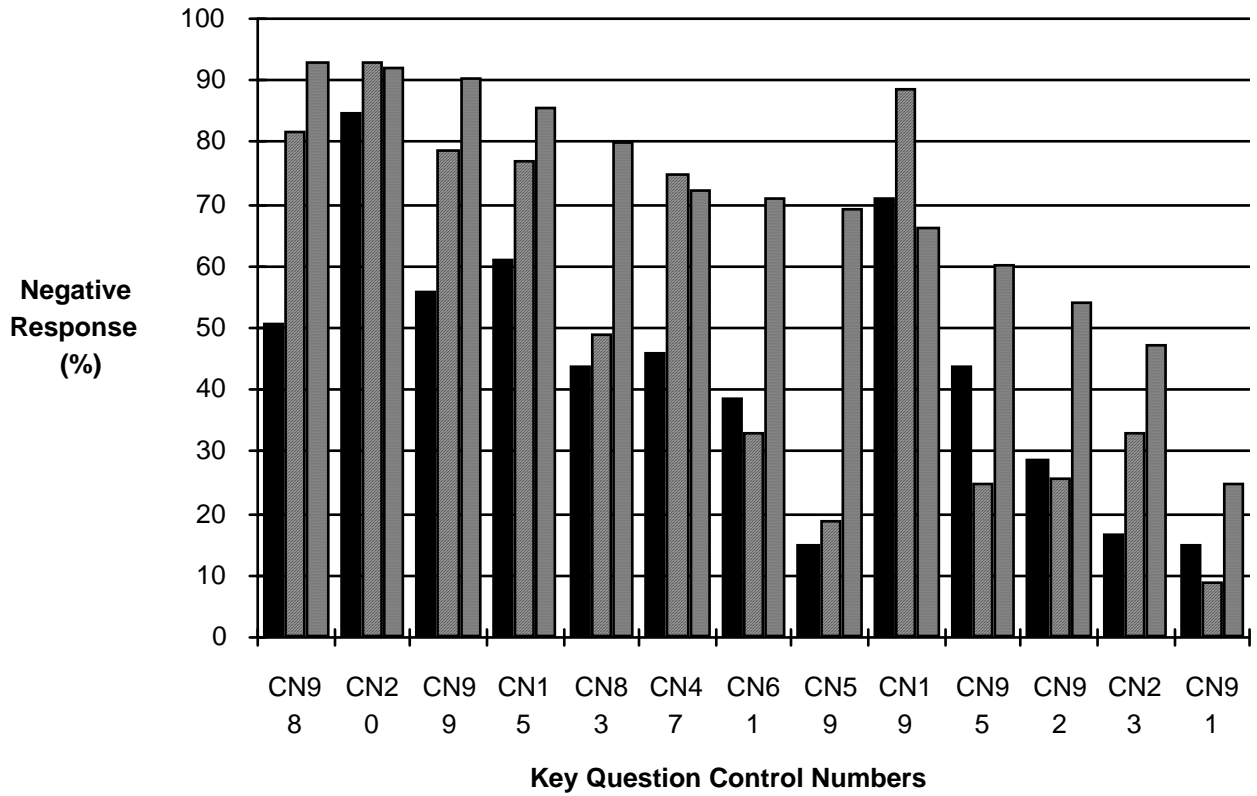
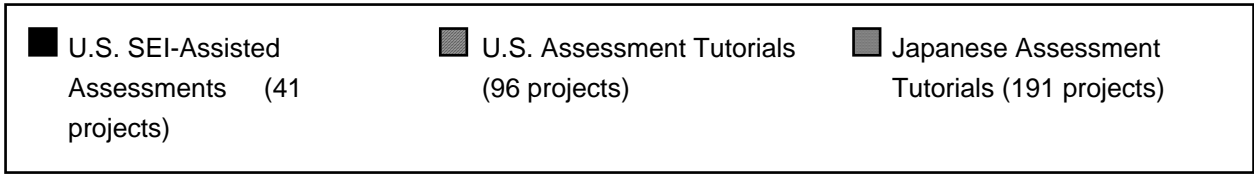


**Figure A.6. Percent Negative Response of Level 1 Projects to Level 3 Key Questions (U.S. SEI-Assisted Assessment Data - 41 Projects)**





**Figure A.7. Comparison of Percent Negative Response of Level 1 Projects to Level 2 Key Questions**



**Figure A.8. Comparison of Percent Negative Response of Level 1 Projects to Level 3 Key Questions**