FIFTY QUESTIONS A CHIEF EXECUTIVE SHOULD ASK ABOUT SOFTWARE

February 25, 2007

Abstract

Software is a critical factor in many aspects of corporate operations. Unlike most aspects of corporate operations, software has been difficult to bring under full executive control. Many chief executive officers (CEO's) have only a limited knowledge about both computers and software. This article discusses 50 key questions that CEO's should ask about, in order to ensure that the software their companies depend upon will be an asset and not a liability to the corporations they control. Although specific answers within specific companies are needed, general answers to the questions are given to illustrate the ranges of possibilities.

Capers Jones, Chief Scientist Emeritus
Software Productivity Research LLC

Phone (401) 789-7662
Email CJonesiii@cs.com

INTRODUCTION

Software has been a difficult technology to bring under executive control. In spite of the importance of software to corporate operations, it has remained an intractable technology, which causes almost as much trouble as it brings benefits.

Chief executive officers (CEO's) of modern corporations are responsible for more software and more software employees than any other occupation group in human history. However, many CEO's are 40 to 60 years of age, and their background and training did not include much information about software and how to deal with it. This is also true of many vice presidents of operating units such as manufacturing, sales, marketing, and human resources.

In the 21st century every major corporation is largely automated for financial reporting, billing, marketing, and often manufacturing as well. Software has become a critical factor for the time-to-market of many new products. For a surprising number of high-technology products, software is actually embedded in the product itself and therefore may be subject to warranties.

This article discusses the 50 of the major software topics that CEO's should know about, to ensure that the software their companies depend upon will be an asset and not a liability to the corporations they control. There are a number of questions that CEO's should ask their Chief Information Officers (CIO's) and other key software executives, and they are divided into five sections:

- Current "hot" topics regarding software
- Software usage, user satisfaction, and value with the corporation
- Employee satisfaction and demographic questions
- The economic impact of software on the corporation
- Competitive analysis of how other corporation's deal with software

Not only should the CEO know the answer to these questions, but so should the CIO, Vice President of Software Engineering, Vice President of Engineering, and any other senior executive with software staffs and responsibilities.

The answers to the questions would normally come from a CIO, Vice President of Information Resources, Vice President of Software Engineering, or from an equivalent position.

However, senior executives themselves would be able to get answers to many of the questions available only if the corporation has a fairly sophisticated software measurement system. The answers also assume that software organizations exist which are chartered to analyze and improve software performance, i.e. one or more of these kinds of organizations are present in the company:

- Software quality assurance
- Software process improvement
- Software measurement and metrics

These three kinds of organizations are usually associated with leading-edge corporations. If such organizations are not present and if the questions are unanswerable within a company, it is a sign that software is not fully under executive control.

Leading-edge companies probably would have the answers to all of these software questions readily at hand, due to annual surveys of software-related topics which leading companies tend to carry out.

Lagging companies, or companies where the CEO has no interest in, nor knowledge of, the significance of software face an uncomfortable future when dealing with more knowledgeable competitors.

For the purposes of this article, the questions are given first, and then the general answers to the questions are discussed afterwards. Readers are urged to put themselves in the position of the CEO of a major corporation, and see how many of these important questions they can answer for their own corporations.

Current "Hot Topic" Questions Involving Software

1. How many of our systems are more than 10 years old and becoming obsolete?
2. How much are we spending on Sarbanes-Oxley each year?
3. How safe are our trade secrets and financial data?
4. How safe are we from viruses, spyware, and hacking?
5. How useful would ERP packages be to us?
6. What is the value of ISO 9000-9004 certification for software?
7. What is the impact of moving up the SEI capability maturity scale?
8. What is our current level on the SEI CMM maturity scale?
9. What are we doing with the Internet and the World Wide Web?
10. What are we doing with Data Quality and Data Warehousing?

Software Usage, Value, and User Satisfaction Questions

1. How many of our employees and managers use computers and software today?
2. What do our people think about the value of our software in doing their jobs?
3. What are the top complaints our people make about our software?
4. Are our customers satisfied or dissatisfied with the software we deliver?
5. Are our customers satisfied or dissatisfied with our service?
6. How many problem reports do we get a year from customers?
7. Are we in danger of being sued because of problems with our software?
8. What are we doing to keep our software useful and valuable?
9. What are our worst current software systems?

10. What are we doing to improve our worst software systems?

Employee Satisfaction and Demographic Questions

1. Do we have the best software people in our industry?
2. What is our current breakdown between software employees and contractors?
3. How many software personnel do we employ this year?
4. How many will we need five years in the future?
5. Are there any critical skill shortages that we're having trouble recruiting?
6. Are our software personnel satisfied or dissatisfied with us?
7. Is our voluntary attrition rate higher or lower than our competition?
8. How do we stack up to the competition in terms of benefits and salaries?
9. What advantages or disadvantages would we get from outsourcing our software?
10. What are the pros and cons of offshore outsourcing?

Software Economic Impact Questions

1. How much software do we own right now?
2. What is the replacement cost of the software that we own?
3. What is the taxable value of the software that we own?
4. What is our annual growth rate in software ownership?
5. How many of our software projects are canceled and don't get finished?
6. What are the main reasons for our canceled projects?
7. How much have we spent on canceled projects over the last five years?
8. How much have we spent on software warranty repairs over the last five years?
9. How many of our applications need immediate replacement?
10. How big is our backlog of applications we can't get built?

Competitive Analysis Questions

1. Which companies in our industry are best in software?
2. Which companies should we benchmark with?
3. If we spent more on software, what should we spend it on?
4. If we spent less on software, what harm would it do to us?
5. How does our software productivity compare to our competition?
6. How many of our software projects run late and exceed their budgets?
7. How does our software quality compare to our competition?
8. What are our main strengths in the software domain?
9. What are our main weaknesses in software?
10. Are we training our people so they stay current with new technologies?

ANSWERS TO THE FIFTY EXECUTIVE QUESTIONS ABOUT SOFTWARE

The following general answers to the 50 key software questions are taken from among our client base. Of course, what is important for any specific company is not general knowledge, but how that company in particular would answer the questions.

The ranges of possible answers are very broad, and vary widely by industry and also by company size. Insurance and banking, for example, were early adopters of automation and tend to have large populations of software professionals. Other industries, and in particular conglomerates, may have answers that vary by location and business function.

Nonetheless, software is an important enough management topic so that the senior executives and CEO's of all medium and large enterprises should have a basic understanding of the economic impact of software on corporate operations.

Answers to the Current "Hot Topic" Questions

In 2007 all corporations face a number of expensive and important problems. One of these problems is the fact that most portfolios of software are aging, unstable, and difficult to modify safely.

Another major problem is that every day every corporation has to mount expensive and vigilant defenses against spyware, viruses, and computer hackers. Proprietary information is expensive to safeguard, and often more vulnerable than it should be.

A third major problem that faces large corporations in the Fortune 500 class concerns the high costs of Sarbanes-Oxley compliance, and the legal problems associated with non-compliance. Of course corporations should also be concerned with the underlying issues that triggered Sarbanes-Oxley in the first place: misstatement of financial results and accounting practices that conceal losses and exaggerate profits.

The next "hot topic" concerns the set of related quality standards issued by the International Standards Organization (ISO). These standards are numbered from ISO 9000 through ISO 9004, but the particular standard that has the greatest relevance to software is ISO 9001.

Achieving ISO certification is often necessary to market various products in Europe, including software products. However, empirical data on the effectiveness of achieving ISO certification is sparse. Indeed, as this article is being written an Internet conference is going on in which some participants assert that ISO certification may degrade quality rather than improve it.

The consensus among top software organizations with state of the art quality control is that ISO certification is fairly easy to achieve, but does not add much value to the

measured quality of software products, nor even add much in the way of value to software development processes.

The next "hot" topic concerns the well-known Software Engineering Institute (SEI) which is a non-profit organization funded by the U.S. Department of Defense. The SEI has developed an interesting five-point evaluation scale called the "capability maturity model" or CMM for short.

The SEI CMM scoring system rates companies based on patterns of answers to about 150 questions. The most primitive level, where organizations typically have trouble with software cost and schedule overruns, poor quality, and cancellations is Level 1 which is termed the Initial level using the SEI capability.

As of 2007, about 75% of all organizations are found at this level. The approximate distribution of the five levels is shown in table 1:

Table 2: Distribution of Results on the SEI Capability Maturity Model

CMM Level	Frequency	Meaning
1 = Initial	75.0%	Seriously inadequate methods and tools
2 = Repeatable	15.0%	Some rigor in processes and methods
3 = Defined	7.0%	Good enough to develop reusable materials
4 = Managed	2.5%	Very good in most software topics
5 = Optimizing	0.5%	State of the art in all software topics

The empirical evidence of the benefits of moving up the SEI CMM scale is controversial as of 2007. In general it requires about a year to move from level to level. Ascending to each higher level may cost as much as $5,000 per capita for the entire software population. There is some overlap between the various levels in terms of both quality and productivity. However, as empirical data slowly accumulates, it appears that the higher levels 2, 3, 4, and 5 often have better quality and sometimes better productivity than level 1.

The two final "hot topics" are complex issues that deserve more extensive treatments than can be given in a short article. Both usage of the Internet for business purposes and the consolidation of corporate data into unified "data warehouses" are expanding rapidly in importance.

Every major corporation now has a presence on the Internet and a world-wide web site for marketing and advertising, and often for actual business operations.

The data warehouse concept of consolidating information from various corporate data bases is leading to new kinds of corporate-wide studies that were difficult to perform when data was stored in separate and sometimes incompatible data bases. However, the

data warehouse concept is also focusing attention on the “hot” topic of data quality. Errors in data bases are both common and severe, and successful data warehousing demands a careful scrubbing of data in order to eliminate data errors.

Answers to the Software Usage, Value, and User Satisfaction Questions

Depending upon the industry, from less than 25% to more than 70% of U.S. white collar workers have daily-hands on usage of computers and software. Some of the industries with the highest levels of daily use include insurance, finance, telecommunications, and of course the computer industry itself. Daily usage of software is often inversely correlated with position level: executives may use software less than technical staffs and lower management.

Most software users feel that software helps their job-related tasks, but that software is often difficult to learn and sometimes of less than optimal quality. Based on surveys of commercial-grade software in the U.S. about 70% of users are fairly well satisfied with both quality and service, but 30% are not. For internal MIS applications, about 65% of users are satisfied with quality and service, but 35% are not. Basic complaints include poor quality, slow service, and software that is not easy to use. For internal MIS software, slow development schedules leads the list of complaints.

Starting in about 1985, a new subindustry appeared of companies and products that can provide "geriatric care" for aging software. It is now possible to analyze the structure of existing software and automatically restructure it. Reverse engineering and reengineering are newer technologies for analyzing and converting aging software applications into modern versions.

The leading companies in many industries have started to reverse the alarming growth in software maintenance and enhancement costs by using these techniques. Some industry leaders are now spending less than 20% of their annual software budgets on maintenance and enhancement of aging software, while laggards may be spending as much as 70%. If your company’s maintenance and enhancement costs exceed 50% of your total software budget, something may be seriously wrong in your enterprise.

Application of geriatric technologies to aging software can not only stretch out the useful life of software, but also facilitates eliminating critical problems such as the year 2000 problem.

The area of maintenance has been one of the most successful targets of the outsource community. Maintenance outsource vendors are often almost twice as productive as the companies whose software they are maintaining. This is due in part to having trained maintenance personnel, and in part to having excellent tool sets. Also, maintenance outsource groups are usually not splitting their time between new development projects and maintenance of legacy applications.

Answers to the Employee Satisfaction and Demographic Questions

The growth of software professionals has been extremely large since the industry began in the 1950's. Although the growth rate is now declining, it is still positive: the overall growth of software professionals in the U.S. appears to be rising at a rate of 2% to 5% per year. This is down from the 13% of the early 1980's, and the 20% of the 1970's.

The percentage that software professionals constitute of total employment varies with industry: in banking and insurance, from 7% to 15% of all employees can be in the software functions. Large banks and insurance companies, for example, can employee as many as 5000 software professionals in a single location, and they may have more than one such location! Overall, software professionals in the U.S. may approximate 2% of the total work force.

Many other kinds of companies also have high software populations: telephone operating companies, manufacturing, and defense for example.

There are now more than 75 different occupation groups associated with software in large corporations and government agencies: programmers, analysts, quality assurance specialists, data base administrators, technical writers, measurement specialists, estimating specialists, human factors specialists, and many more are now part of software functions.

If your total software staff is larger than 1000 professionals, and your company cannot identify the occupation groups employed, then something is wrong in your software and human resource organizations.

Large enterprises find that they achieve the best results from judicious usage of software specialists: maintenance specialists and testing specialists are among the first to be needed, as are data base administrators. Software, like medicine and engineering, is too vast a domain for generalists to be expert in all of its phases.

As of 2007, a number of software specialties have become so popular that shortages exist in the United States. Indeed, for a few of the specialties it may even be necessary to pay "signing bonuses" more or less like professional athletes receive. Some of the software specialties in short supply as of early 2007 include:

- SAP R/3 specialists
- Six-sigma "black belt" specialists
- JAVA and world wide web specialists
- Function point specialists

If your voluntary attrition rate of software professionals is at 3% or higher, there may be something troubling about your software organization. If your company does not carry out annual opinion surveys, have a dual salary plan that provides equivalent

compensation for technical staffs and managers, or have other modern human resource practices in place, you may not be able to attract and keep good professional staffs.

One critical topic which every executive should have current data on is that of the numbers and costs of canceled software projects. The kind of canceled projects that are of concern are those due to excessive cost and schedule overruns, which are severe enough to reduce the value of the software to negative levels.

There are of course many business reasons for canceling a software project, for example selling a portion of a business, or buying a company that already owns software similar to a project your company is developing. However, most software projects that are terminated are canceled because they are so late and so much more expensive than planned that they are no longer valuable. This topic is perhaps the chief complaint that CEO's have about software: large software projects often run late and overrun their budgets.

Table 3 shows the approximate U.S. distribution of project outcomes, sorted by size into six size plateaus from 1 to 100,000 function points:

Table 3: U.S. Software Project Outcome By Size of Project

	PROBABILITY OF SELECTED SCHEDULE OUTCOMES				
	Early	**On-Time**	**Delayed**	**Canceled**	**Sum**
1FP	14.68%	83.16%	1.92%	0.25%	100.00%
10FP	11.08%	81.25%	5.67%	2.00%	100.00%
100FP	6.06%	74.77%	11.83%	7.33%	100.00%
1000FP	1.24%	60.76%	17.67%	20.33%	100.00%
10000FP	0.14%	28.03%	23.83%	48.00%	100.00%
100000FP	0.00%	13.67%	21.33%	65.00%	100.00%
Average	5.53%	56.94%	13.71%	23.82%	100.00%

Because software is important but has been difficult to control, an increasing number of companies are evaluating outsourcing, or turning over software development and maintenance to companies that specialize in software.

While outsourcing may be advantageous, not every outsource agreement is satisfactory. The approximate distribution of domestic outsource results after two years is shown in table 4:

Table 4: Approximate Distribution of U.S. Outsource Results After 24 Months

Results	**Percent of Outsource Arrangements**

Both parties generally satisfied	70%
Some dissatisfaction by client or vendor	15%
Dissolution of agreement planned	10%
Litigation between client and contractor probable	4%
Litigation between client and contractor in progress	1%

From process assessments performed within several large outsource companies, and analysis of projects produced by outsource vendors, our data indicates slightly better than average productivity and quality control approaches when compared to the companies and industries who engaged the outsource vendors. These results are for development projects. For maintenance projects, the data indicates that maintenance outsourcing is between 25% and 100% more productive.

However, our data is still preliminary and needs refinement as of 2007. Our main commissioned research in the outsource community has been with clients of the largest outsource vendors in the United States such as EDS, Perot, Lockheed CSC, and others in this class. There are a host of smaller outsource vendors and contractors where we have encountered only a few projects, or sometimes none at all since our clients have not utilized their services.

For international outsourcing, the results are more difficult to assemble because they can vary from country to country and contract to contract. However, outsource agreements with countries such as India, the Ukraine, the Philippines, Russia, and China seem to have more or less the same distribution of results as do the U.S. outsource contracts, although the margin of error is quite high.

Answers to the Software Economic Impact Questions

A large bank or insurance company can expect to own from 500,000 to as many as 3,000,000 function points. A major multi-national manufacturing concern or an international conglomerate can own upwards of 5,000,000 million function points scattered across dozens of software labs and locations.

Even a small manufacturing company with less than 500 total employees can expect to own up to 375,000 function points. (Note: function points are becoming the de facto standard for measuring software. A function point is a synthetic metric composed of the weighted totals of the inputs, outputs, inquiries, interfaces, and files which users have identified as significant in a software application.) The growth rate of corporate portfolios is an annual increase of 5% to 7% in new and changed function points.

A typical distribution in a manufacturing company would be to build about 40% of the function points they use, lease about 30%, and buy about 30%. The purchased functions are often personal computer applications, while the leased and developed applications are often for mainframes or mini computers.

Banks and insurance companies may lease or purchase more than 50% of the function points they utilize, and build less than 30%. This is because of the ready availability of commercial banking and insurance packages. Smaller companies buy or lease much more software than they build: very small companies may purchase 100% of all their software functions.

Also, many companies are migrating toward integrated software solutions such as the well-known SAP R/3 software package which can replace more than 50% of in-house applications within many companies.

Replacement costs vary significantly, but a rough rule of thumb for the United States is that each function point would cost from $200 to $3,500 to replace and the 1997 average is roughly $1,200.

The less expensive replacements would be the simpler MIS applications, and the more expensive ones would be complex systems applications that have many special characteristics.

The life expectancy of software is proportional to its volume: systems of more than 5000 function points in size will run for 10 to 15 years in many cases. Therefore a rational replacement strategy must be based on not only the future business of your company, but on the size and class variances of your existing portfolio. The long life expectancy of major software applications is a key factor that explains why the year 2000 problem is so serious and will be so expensive.

Answers to the Competitive Analysis Questions

Software expenses in the U.S. today constitute about 3% to 15% of the sales volumes of typical enterprises. There are major variations by industry, however. Obviously for software companies such as Microsoft, Computer Associates, Oracle and the like software expenses approach 100% of their sales volumes.

For non-software industries, insurance and banking often have the largest expenses of general businesses, although telecommunications companies, defense companies, and computer manufacturers also have the high percentages of software expenses. Both the manufacturing and service companies may be software-intensive, so many businesses will also have a high percentage of software costs.

In data collected from Japan, Europe, and the U.S. it appears that the optimal level of capital investment per software engineer will be about $5,000 to $10,000 for a powerful work station, but even more for software tools and support: $25,000 in round numbers. Historically, software development has been undercapitalized. For software, like all other industries, there is a direct correlation between capital investment and overall productivity.

One of the most surprising recent findings about software productivity is that there is a strong correlation between office space and overall performance. Here in the U.S., software professionals with more than 78 square feet of noise-free office space are in the high quartile in overall performance, while those with less than 44 square feet of open office space or crowded, noisy cubicles are in the low quartile in overall performance.

U.S. productivity for MIS software averages about 8 function points per staff month. In general, large projects of more than 1000 function points will have much lower rates: in the range of 1 to 5 function points per staff month.

Small projects of less than 100 function points in size can often exceed rates of 20 function points per staff month. The overall range of software in the U.S. is from a fraction of a function point per staff month to a high of about 140 function points per staff month. Systems software and military software averages are much lower than MIS averages, due to the greater complexity of those applications and, in the case of military software, the enormous volume of paperwork required by military specifications.

U.S. software quality for MIS projects is an average of about 4 defects per function point, with perhaps 75% of those defects being found prior to delivery of the software to its users. An average of about 1 defect per function point can still be present in MIS software at delivery.

Systems software and military software both have higher defect potentials, but also higher defect removal efficiencies: systems software averages more than 90% in defect removal, and military software can average more than 95%. To achieve an excellent reputation for high software quality, strive to keep potential defects below 2.5 per function point, and to achieve defect removal efficiencies in excess of 97%, for a delivered defect rate of less than 0.075 defects per function point.

There is a strong correlation between high levels of user satisfaction and low levels of delivered defects, which might be expected. However, there is also a strong and surprising correlation between low levels of delivered defects and short development schedules. Since finding a fixing bugs is the most expensive task in all of software, leading-edge companies that utilize sophisticated combinations of defect prevention and defect removal techniques can achieve very short schedules and very high quality levels simultaneously!

Just as a medical doctor cannot tell a specific patient about medical problems without a diagnosis, it is not possible to tell a CEO about specific strengths or weaknesses without careful study. However, common software strengths found in the United States include experience levels and capabilities of the staff, adequacy of basic tools, use of high-level languages, and use of structured methods for development and enhancements.

Common software weaknesses in the United States include excessive schedule pressure, management skills that rank below technical staff skills, failure to use adequate pre-test reviews and inspections, and generally lower levels of capital equipment and support than appears desirable.

A number of software research and consulting organizations specialize in performing software diagnostic studies: for example the Gartner Group, Software Productivity Research (SPR), and the Software Engineering Institute (SEI) and a number of other organizations can carry out software assessments and/or benchmark studies.

Now that the questions and answers have been stated, there is one significant final point: if your company has a large software population, but answers to these questions are difficult or impossible to ascertain, you may need to consider either outsourcing, or a reengineering of your software organization so that topics such as the ones in these questions can be dealt with. You may even need new executive-level positions reporting directly to the CEO and charged with leading software functions into the modern era, such as creation of a Vice President of Software Engineering or a Vice President of Quality Assurance.