DATA QUALITY AND DATA METRICS

The topic of data quality was not included in the original 1992 SPR quality survey from which this book derives. However, data quality is a topic of growing significance as more and more of the world's important records are stored in computerized form. Data quality is also very difficult to measure and is outside the scope of normal software quality assurance since much of the data comes from clients using the application long after development rather than from the software development process itself.

Among SPR's clients almost all of them (about 600) have data quality problems, but only about 75 have begun any kind of formal data quality improvement programs. The normal trigger for data quality improvement is the desire to move toward data warehousing, data mining, online analytical processing (OLAP), and other approaches where data from various sources needs to be consolidated. As a result, about another 50 of SPR's clients are discussing data quality and will probably move toward formal data quality approaches in 1997.

Data and databases can have any of the same four kinds of kinds of defects discussed earlier in the section of this book dealing with error categories:

- Errors of omission (something missing, such as the zip code in an address)
- Errors of commission (something incorrect, such as a misspelled name)
- Errors of clarity and ambiguity (primarily with complex search or join criteria)
- Errors of speed or performance (primarily sluggish transactions with large databases)

Unlike software, data has another kind of error condition too. Data errors can occur due to changes in the real-world situation which the data describes or models. Thus records that are correct at a specific time and date can become erroneous as time passes without any change in the record itself but because of changes in the reality which the data describes.

For example, suppose on January 1 of 1997 a corporate personnel database has records for an employee named Jane Doe, who is employed as a database administrator, with a salary of $50,000 per year.

Now suppose that on February 1 of the same year the employee marries and changes her name. On March 1 of the same year the employee is promoted and becomes a "senior database administrator" with a salary of $60,000 per year.

Unless these real-world changes are reflected in the data describing the employee, the records will be no longer be correct although they were correct when the data was first put into the database.

For data errors brought about by gradual changes in the real, external world which the data models, there is no easy solution. Active and frequent comparisons between the data and reality must be performed.

These comparisons may require voluntary participation on the part of humans and companies whose information is contained in the databases. As a trivial but very common example, when a person moves or changes addresses, it is up to that person to notify at least the post office in order to continue to receive mail.

However, most people end up notifying a large number of vendors, journals, credit card companies, and professional service groups such as doctors, dentists, attorneys, and accountants. What these people are really doing is participating in a number of database update transactions, even if they are not consciously aware of it.

Another and very serious example of a situation where reality will change and cause data errors is the approaching "Year 2000" problem. Not only do millions of software applications contain two-digit date fields which will fail when 1999 becomes 2000 AD, but so do databases, repositories, and data warehouses.

Since many companies often own larger volumes of data than they own software, the Year 2000 problem will generate tremendous data quality problems as well as tremendous software repair problems, although the data quality aspects of the Year 2000 issue are not as well understood.

Data has been a difficult topic to perform research on. A major source of difficulty has been the lack of useful data-related metrics. Unfortunately, as this draft is written in 1997 there are no satisfactory normalizing metrics for expressing the volume of data a company uses, the quality levels of the data, and the costs of creating data, migrating data from platform to platform, correcting errors, or destroying obsolete data.

Also, unlike software errors which are found only in computerized applications, data errors are not restricted to data stored magnetically or optically in computerized databases, repositories, and warehouses. Data errors can also occur in data stored on paper, on microfiche, on video or audio tape, on CD-ROM, or on any other known medium. Indeed, data errors are far older than the computer era and have been occurring since the invention of writing and symbolic representations for mathematics.

There are many data errors to be found in ordinary office file cabinets containing paper documents. Data quality is also of concern during transmission of data via optical or copper cable, microwave, or by radio or television signals since errors and noise can be introduced during the transmission process.

Even the word "data" itself is ambiguous and hard to pin down exactly. What the word "data" means in the context of data quality are symbolic representations of facts (i.e., pi = 3.14159265358, etc.), encoded representations of objects (i.e., valves, bolts, chairs, etc.) or encoded representation almost any kind of idea the human mind can envision.

Data is not necessarily static, and indeed some kinds of data can change very rapidly. For the data on the actual and predicted trajectory of an incoming Exocet missile picked up by a ship's radar can change very rapidly indeed.

Associated with the definition of data is the concept that data can be structured in the form of models that deal with various aspects of the world: physical objects, business rules, mathematics, images, and so forth.

Although the function point metric was created to measure the size of software applications, it is interesting to evaluate whether function points might be extended to deal with the size and quality of databases, or whether an equivalent "data point" metric might be developed for similar purposes.

Recall that the function point metric is comprised of the weighted totals of five external aspects of software applications:

• Inputs	Forms, screens, sensor-based values etc., entering the application
• Outputs	Reports, screens, electronic signals etc., leaving the application
• Inquires	Question/answer pairs to which the application responds
• Logical files	Record sets maintained within or by the application
• Interfaces	Record sets passed to or received by external applications

These five elements are then adjusted for complexity to provide the final total of function points for the application. There are 14 adjustment factors for the U.S. function point defined by the International Function Point Users Group (IFPUG) and 19 adjustment factors for the British Mark II function point.

While all five of the function point elements overlap data to a certain degree, they do not seem to capture some of the topics of concern when exploring data quality. Perhaps a set of factors similar to these may move toward the development of a data point metric:

- Logical files The number of record sets maintained within or by the application
- Entities The number of kinds of objects within the database
- Attributes The number of qualifications for the entities within the database
- Inquires Question/answer pairs which the database responds to
- Interfaces Record sets passed to or received by the database

While the above is only a preliminary suggestion, there is an urgent need to develop a data point metric without which research into data quality and database economics are severely handicapped.

Right now, there is so little empirical, quantified information on data quality that this topic is included primarily to serve as a warning flag that an important domain is emerging, and substantial research is urgently needed.

Several commercial companies and Dr. Richard Wang's data quality research group at MIT have begun to address the topic of data quality. Dr. Wang and his colleagues are definitely broadening the horizon of the data quality domain. Indeed the MIT data quality research group is the sponsor of perhaps of the first international conference on data quality in October of 1996.

In spite of good preliminary research, there is still a long way to go before data quality becomes a well-understood topic. To date, there are no known published statistical studies that include the volumes of data errors in either paper or computerized data bases, the severity levels of data defects, or the costs of removing data errors. Even reliable information on the sizes of data bases is difficult to come by, as is corollary information on the costs of collecting the data, validating it, or removing outdated information.

Yet another of the unknowns and ambiguities of the database domain is that of extraneous information and redundancy. There is a tendency to store information that may never be used again, on the grounds that "it is there if someone needs it." It would be very interesting to do a statistical analysis of a full corporate database or repository in order to ascertain how much data is used daily, weekly, monthly, annually, rarely, or not at all. Very preliminary observations indicate that perhaps as much as half of the entire volume of data might fall under the "rarely used" or "never used" categories.

Redundancy in data is also a troubling topic, although better covered in the literature that extraneous data. Most corporations maintain multiple copies of a significant portion of their data. Sometimes the copies are maintained for reasons of security or disaster recovery. Sometimes the copies are maintained for reasons of transaction processing efficiency.

However, some redundancy has no rational explanation and is difficult to explain except under the hypothesis that few companies actually know what kind of data they store, so redundancy is a natural byproduct of partial data dictionaries or catalogs of stored information. Another reason for redundancy between paper data and computerized data is the lack of simple, portable reading devices that would make access to online data easy and facile.

In terms of data quality, anecdotal reports on both manual and computerized databases indicate that errors are both plentiful and severe. Samples of things like employee personnel records, credit records, and mailing lists tend to indicate that errors can occur more than 1% of the records stored. However, much more research is needed to examine the interaction of data errors with the nature and volume of the data itself.

As this book is written, there is not even any accepted standard method for exploring the "cost of quality" for data errors in the absence of any normalizing metrics. This is an important research project and all technical contributions would be welcome.

A cost structure that might be suited to the nature of the database domain would resemble the following, which is similar to the expanded cost structure discussed in the Cost of Quality section earlier in this book.

Costs of Ensuring Data Quality

1) Data defect prevention costs

2) User satisfaction data optimization costs

3) Data quality defect prevention costs

4) Data quality defect removal costs

5) Data quality awareness/training costs

6) Non-test data defect removal costs (reviews, inspections, walkthroughs, etc.)

7) Testing data defect removal costs (all forms)

8) Data-related customer support costs

9) Data-related warranty support and product recall costs

10) Data-related litigation and damage award costs

11) Savings from reduced data scrap/rework

12) Savings from reduced data-related computer downtime

13) Data quality value from reduced time-to-market intervals

14) Data quality value from enhanced competitiveness

15) Data quality value from enhanced employee morale

16) Data quality return on investment (ROI)

Here, too, this is a preliminary list of cost elements that are being presented to elicit further research on a topic of significant economic consequence.

Database and Information Usage Within Large Corporations

I worked for IBM for a 12 year period. During the latter part of this period, I performed a study on the volumes of information required to create IBM's major software products. Although the study did not cover other kinds and uses of information, such as hardware manufacturing, marketing, or sales, it was obvious that information was a major component of IBM's annual expenses and the company utilized enormous volumes of information. The same observation has held true when consulting with other corporations and government agencies: both data and information are major but largely invisible components of operating expense.

The following is an attempt to put together a rough estimate of the volume of data or information typically owned by major corporations. Assume a high-technology manufacturing corporation with a total employment of 250,000 personnel. Roughly 150,000 of those personnel might be engaged in manufacturing activities, while 100,000 would be engaged in various business and operational activities such as engineering, marketing, sales, finance, human resources, administration, purchasing, and the like.

The total number of discrete products and services which corporations of this size typically have on the market would amount to perhaps 5,000 different hardware products and perhaps 500 different software products. Information about these products would be a major component of the company's databases, repositories, and data warehouses.

In addition, a company of this size would have created more than 1,500 internal software tools for use within the company, and would have bought or leased an equivalent number from outside vendors. Both hardware and software would be supported and surrounded by various kinds of sales, marketing, training, repair, and maintenance services. (Although these tools are important and were expensive to acquire or build, it is interesting that even IBM did not have a complete inventory or database of all tools owned or leased by the corporation. This is also true of my other corporate and government clients.)

A corporation of this size would typically have a client base of perhaps 500,000 corporations and government agencies on a global basis. Here too, this is a major kind of information that would be stored in corporate data bases and repositories.

Since many of the client organizations would be large enterprises in their own right, the total number of customer sites served would be in the range of 2,000,000. The total number of real people that are clients or prospects of a company this size would probably approach or exceed 10,000,000 on a global basis.

These three statistics are significant in calculating the volume of information a company maintains as permanent records to conduct basic business operations. Information on personnel, products, and clients constitute the basic operating information that drives modern businesses. The table shows approximate volumes of information stored by a large corporation for these three major kinds of business data might total to the following:

Table 1 Volumes of Primary Business Information Used by A Large Corporation

Information Item	Number of Pages	Number of Words
Personnel Information	12,500,000	3,500,000,000
Product Information	26,500,000	7,950,000,000
Customer Information	65,000,000	19,500,000,000
TOTAL	*104,000,000*	*30,955,000,000*

The latter two kinds of information, on products and customers, have close logical ties and often need to be coupled together for business operations. Indeed, relating clients and products is a major form of transaction for database software in all corporations.

Personnel and staff information, on the other hand, is usually kept rigorously separate from other kinds of information for reasons of confidentiality. Usually it is not possible to even determine the identities of the employees who worked on a product, except in the case of some kind of litigation where the records are produced due to a court order.

Although the three basic kinds of business information constitute, by volume, the bulk of the information which corporations store and utilize in order to conduct business, there are many other kinds of information utilized in the course of business activities.

From studies carried out by the author on technical specifications and user documentation for software products, roughly 30% of the pages contained some kind of graphical illustration or image. Marketing materials and training materials approach 50% graphics and illustrations by volume: that is, every other page contains an illustration. Assuming these ratios hold for hardware products as well as software (hardware was not studied by the author) then the approximate volume of images and graphical information within the case study company would amount to about 10,000,000 discrete images or graphical illustrations.

Although not "pages" in the traditional sense, the volume of data required to encode a typical image or graphic device is much larger than a page of text. However, for convenience in expressing the data in this article, each image or graphic will be deemed equivalent to "one page." It is reasonable to assume about a 50% split between paper and online storage for graphics and images.

One of the data warehouse topics needing additional research is a better survey of the ratios of graphics, images, text, and tabular information utilized by enterprises. Since graphics and images are more troublesome to store and transmit electronically, and since many commercial databases are cumbersome for images and graphics, this is a topic needing substantial future research.

Among military and scientific communities, data can also consist of various kinds of electronic signals from sensors or bit streams coming in from ships, aircraft, satellites, etc. These data items tend to be voluminous and also have the characteristic that they may represent dynamic phenomena in rapid motion.

However, the following case study deals only with "ordinary" business data in the form of alphanumeric information plus standard graphics and some photographic illustrations. Let us now consider three views of information storage and usage in large corporations:

1) The total volume of information stored

2) The volume of information kept in paper form

3) The volume of information kept in magnetic or optically encoded form

Note that there is a high margin of error in this kind of analysis, and the approximations for the ratios of paper to online storage are only hypothetical. An important point for data quality purposes is that errors in paper-based data can be just as severe as errors in computerized data.

The following table summarizes the approximate overall volumes of information in rank order:

Table 2 Relative Volumes of Stored Information for Case Study Example

Kind of Information	Pages Stored	Percent Stored Online
Customer information	90,000,000	50%
Product information	50,000,000	50%
Software applications	40,000,000	75%
E-mail messages	30,000,000	95%
Reference information	15,000,000	20%
Personnel information	12,500,000	50%
Graphics/images	10,000,000	50%
Correspondence	5,000,000	10%
Defect information	2,500,000	50%
Supplier information	1,500,000	25%
Tutorial/training material	1,000,000	50%
Litigation/legal information	1,000,000	25%
Total Volume	*272,000,000*	

Since the case study corporation was stated to have 250,000 employees, it is interesting to note that the total volume of corporate information stored by the case study corporation amounts to more than 1,000 pages per employee. Let us now turn to the volumes of paper information, once again shown in declining order by volume.

Table 3 Relative Volumes of Stored Paper Information for Case Study Example

Kind of Information	Pages Stored
Customer information	45,000,000
Product information	25,000,000
Reference information	12,000,000
Software applications	10,000,000
Personnel information	6,250,000
Graphics/images	5,000,000
Correspondence	4,500,000
E-mail messages (printed)	1,500,000
Defect information	1,250,000
Supplier information	1,125,000
Litigation/legal information	750,000
Tutorial/training information	500,000
Total Volume	*112,875,000*

To give a human context to the volumes of paper information, consider that 250 pages of ordinary 20-pound office paper make a stack one-inch high. The volume of paper information stored by the case study corporation is roughly equal to a stack of paper 37,625 feet high, or more than seven miles of paper information. Expressed another way, if the paper information were divided equally among each employee, then about 451 pages would be assigned to every worker. If the 2,250,000,000 pages of personal reference information stored privately by employees is considered, then it would constitute a stack 750,000 feet high, or more than 142 miles.

This is equivalent to about 9,000 pages for every employee. For example, my office at Software Productivity Research contains one wall of shelves that are six feet high and seven feet wide. The shelves contain almost 40 linear feet of books, reports, and data from our own and secondary sources.

In addition, I have about six cubic feet of lockable storage in a file cabinet, and another dozen or so cubic feet reserved for my data in our main file room. However, the file cabinets contain corporate data and client studies. It is interesting that I more more private data in my shelves than corporate or business-related data. Most of the SPR consultants, managers, and technical staff have similar arrangements with books and articles that they find interesting and relevant.

Currently no known data warehouse could absorb this volume of personal and private reference, so it is fortunate that much of this kind of information is treated as personal property and not as a corporate asset. Eventually, however, for data warehousing to become a truly effective business technology this private information must be absorbed.

Let us now consider the volumes of information stored online in magnetic or optical formats.

Table 4 Relative Volumes of Online Information for Case Study Example

Kind of Information	Pages Stored
Customer information	45,000,000
Software applications	30,000,000
E-mail messages (online)	28,500,000
Product information	25,000,000
Personnel information	6,250,000
Graphics/images	5,000,000
Reference information	3,000,000
Defect information	1,250,000
Tutorial/training material	500,000
Correspondence	500,000
Supplier information	375,000
Litigation/legal information	275,000
Total Volume	*145,625,000*

Since this case study reflects a large and sophisticated corporation, it is interesting to note that even though more than 53% of the total volume of business information is available online in magnetic or optical form, there is still almost 47% of the total information is stored in paper form.

Moreover, if the 2,250,000,000 pages of private reference information kept in individual offices in the form of paper is considered, the total volume of online information for the case study company is less than 6%.

Also significant is the redundancy between paper and online storage. For several kinds of information necessary to business operations, such as basic product and customer information, training, and tutorial information, there is almost 100% redundancy between paper and online storage of information.

There is also an approximate 100% redundancy between paper and online storage for many kinds of financial and accounting data, and for many aspects of the data associated with litigation. In these two situations, various government audit trail requirements or court rulings on the admissibility of evidence are among the reasons for the redundant storage.

However, the dominant reasons for redundancy between paper and online information storage are convenience and accessibility. It can by hypothesized that much of the redundancy between online and paper forms of information can be traced to the lack of convenient portable access devices for extracting and utilizing online information.

If hand-held, portable devices were available which had screens equivalent in clarity to printed pages, and were about as easy to carry and use as normal books, then the need for printed information should decline significantly. This kind of device is technically achievable and indeed products such as the SONY Data Diskman are steps in the right direction.

If the amount of information in the case study example is multiplied by all the major corporations and government agencies of the world, it can be seen that huge volumes of information are a major contributor to a number of escalating social problems such as rising taxes and increasing health-care costs.

It can be hypothesized that a major portion of global tax dollars and medical costs are tied up in the production and storage of enormous volumes of words and diagrams, rather than the provision of actual services.

Error Densities of Stored Information

Data quality, or the prevention and elimination of errors in databases and data warehouses, is notoriously difficult to quantify. As already stated, data quality is perceived to be an important topic but there is a severe shortage of empirical data on the volumes and severity levels of data errors. There is also a lack of fundamental metrics for normalizing data quality.

The current U.S. averages for software defect potentials is a total of about 5 bugs or errors per function point, coupled with an approximate 85% defect removal efficiency so that at deployment the volume of latent errors is about 0.75 defects per function point.

Assuming that a hypothetical "data point" metric existed that approximated software function points for sizing purposes, the error densities of data errors appears to be much higher, and the defect removal efficiency levels much lower, than for software.

From analysis of errors in the SPR client databases and repositories, I hypothesize that the data defect potential for the United States is around 8 errors per "data point" and the removal efficiency is only about 75%, so that the number of data errors deployed would be about 2 per "data point" or more than twice the volume of software errors at delivery.

This assertion poor of data quality measured with hypothetical "data point" metrics is of course purely speculative. However, from thoughtful comparisons of data quality and software quality, it does appear that the error densities associated with data may be high, and everyone knows that it is extremely difficult to remove data errors. What is needed, of course, are exact metrics that can capture both the data quality defect potentials and the data quality defect removal efficiency levels. Without quantification of these two factors, data quality research is little more than subjective opinion.

In the context of this case study, my observations of printed reference manuals and tutorial information indicates a rough "defect potential" of about three errors per page of information when it is initially created.

Normal proof reading and fact checking of textual information is fairly efficient. Studies of magazines and book publishers indicate that careful proof reading by a single proof reader can eliminate about 90% of minor typographical errors, and the use of two pairs of proof readers can top 99%.

However, there are no equivalent studies for ordinary business databases. However, based on the number of errors reported the efficiency appears lower. From my preliminary analysis, assume that out of three errors per page fact checking might eliminate 2.75 of them, so there would be roughly 0.25 errors per page still latent in information when it is stored in a database or one error for every four pages of stored data.

Assuming that these preliminary results are true for the entire mass of information considered as part of this case study, then the total number of potential errors in the data would have amounted to 816,000,000. Assuming that roughly 92% of them would have been eliminated, the total volume of latent errors still remaining would total to about 65,280,000. This is a significant volume of latent errors.

Even if only 2% of the latent data errors are important enough to cause serious damages (i.e., errors in tax information, errors in payroll information, errors in customer orders, errors in shipping address, etc.) that is still more than 1,000,000 data errors, which is quite a significant volume. The twin topics of data errors and data quality are obviously very important, and deserve serious research.

The literature on database design, data warehousing, and the "information superhighway" tend to deal with only portions of the kinds of information recorded and utilized by corporations and government agencies. There is a growing need for careful exploration and thoughtful analysis of all of the kinds of information stored in paper from, magnetic or optical form, or stored redundantly in multiple forms.

This hypothetical case study attempts to construct a profile of all of the kinds of information recorded and used by major corporations in carrying out their business and personnel operations.

Although hypothetical, the case study strongly suggests that several topics are in need of significant future research in order to develop truly effective data warehouse and data quality concepts and tools:

- The need for an effective "data point" metric similar to function points for software is a critical gap. Without effective normalizing metrics for size and quality, database research, data quality research, and all derivative fields of research are severely handicapped.

- Future research is also needed in the related topics of information creation costs, information defect repair costs, information storage and transmission costs, and the almost totally unexplored topic of the costs of eliminating or destroying unwanted and obsolete information.

- The topic of the current near 100% redundancy between paper and online storage for some information deserves continued study. (The main reasons for the redundancy appear to be security, performance, the lack of light-weight hand-held reading devices, and lack of effective catalogs that describe information already available.)

- The topic of the ratios of graphics and images to alphanumeric information needs additional study. Currently graphics are more or less ignored in standard database products.

- The topic of sensor-based data also needs to be included in data quality research. Otherwise data coming in from satellites, radar, or electronic sensing devices such as those controlling aircraft flight systems cannot be evaluated under data quality criteria.

- The topic of the enormous volumes of personal and private reference information stored in individual offices, which collectively are far larger than any other kind of data storage, and far larger than any known data warehouse, is essentially unexplored and is not part of any current studies on data warehousing or repositories.
- The topic of the "Year 2000 problem" in databases, repositories, and data warehouses needs urgent research. It is suspected that the problem is as severe for databases as for software, but much less has been published on the data issues of the Year 2000 problem.
- One of the emerging and very serious problems with data storage is long-range retention. I remember reading an article recently where I said that if archeologists dug up the ruins of our civilization in 1,000 years and found CD-ROMS, magnetic disks, magnetic tape, and books, only the books would probably be decipherable. The problem is not so much that the physical media decays, but rather that there is no really long-range format for encoding data that is universal and has longevity.

As computers expand in usage throughout the worlds of business, government, and military operations data and data quality will be growing in importance in the 21st century. Data quality research is only in start-up mode as this book is written, and needs to mature into a fully developed subdiscipline of quality control.

DEFECT DISCOVERY RATES BY CUSTOMERS AND USERS

When software is delivered to customers or users, bugs and defect start being reported back to the development organization. However, the discovery of bugs by users is not instantaneous. The following table shows typical results for several years in a row, with the data showing the percentage of the original delivered defects reported.

Table 1 Three Year Discovery Rate of Initial Software Defects After Release

	End-User	MIS	Outsource	Systems	Commerc.	Military	Average
Year 1	80.00%	30.00%	40.00%	60.00%	65.00%	70.00%	57.50%
Year 2	15.00%	35.00%	35.00%	30.00%	25.00%	25.00%	27.50%
Year 3	5.00%	25.00%	20.00%	9.00%	9.00%	4.00%	12.00%
Latent	0.00%	10.00%	5.00%	1.00%	1.00%	1.00%	3.00%
Total	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%

Of course long before three years have passed new defects will be injected due to updates, enhancements, bad fixes, and the like. However this table shows only the decline in the original defects from release 1. The data is derived from studies performed by the QA teams in IBM, ITT, and also several other SPR clients.

As a general rule software that controls physical devices such as systems software and military software will discover bugs more rapidly than software that deals only with information. The early discovery of software bugs that affect hardware devices is due to the nature of such products. They don't work without reliable software, and if they fail the consequences may be grave.

Indeed, hardware manufacturers such as telecommunication companies tend to have special testing laboratories where the hardware and software are both installed and used by clients under conditions that replicate actual usage patterns after deployment. These special testing labs are designed to ensure that critical problems will be found early.

There are two general rules that determine the rate at which users of software find and report bugs:

- Rule 1: Defect discovery is directly related to the number of users.
- Rule 2: Defect discovery is inversely related to the number of defects.

The first rule is simple and intuitive. The second rule is counter intuitive and needs some explanation.

For the first rule, defect discovery correlates strongly with usage patterns and also with hours of execution. The more customers running the software, the greater the variety of usage patterns and the greater the amount of execution, so defect discovery rates accelerate.

However, end-user software is something of an exception since there is usually only a single user. The reason for the anomaly is because end-user applications are so small (averaging only about 10 function points). Hence the user tends to find all of the bugs in the application fairly quickly. It is much easier to find bugs in a 10 function point application than in a 10,000 function point application.

However, if you ship software with too many bugs in it, say more than one latent defect per function point or seven latent defects per KLOC, the quality and reliability of the application will be so bad that customer usage will probably stop. Zero users report zero defects, and the product will probably be recalled or withdrawn from service.

DEFECT PREVENTION METHODS

A host of technologies have some effect in defect prevention, or reducing the probable number of errors which might occur in software projects. There are too many to discuss all of them, but a few of the most successful defect prevention methods include: Joint Application Design (JAD), prototyping, and various flavors of reusability such as reusable designs and reusable source code.

Among SPR's clients only about 40 companies out of 600 have some kind of formal research programs on-going to explore methods of data prevention. The other companies recognize the importance of defect prevention and utilize quite a few preventive approaches, but they do not actually attempt to explore the impact of prevention on defect totals.

It is an interesting fact that formal design and code inspections, which are currently the most effective defect removal technique, also have a major role in defect prevention. Programmers and designers who participate in reviews and inspections tend to avoid making the mistakes which were noted during the inspection sessions.

It is obvious on theoretical grounds that object-oriented (OO) analysis and design should be an effective defect prevention mechanism. Unfortunately, as of 1996 there is no empirical evidence to support the theory, and indeed there is some evidence that the steep learning curve associated with OO analysis and design may result in higher than normal front-end defect levels for at least the initial projects that use OO approaches.

Following are observations on various flavors of defect prevention and what kind of software they might be appropriate for, or not, as the case may be.

Table 1 Software Defect Prevention Approaches by Application Type

Defect Prevention Approach	Software Classes With Good Results	Software Classes With Questionable Results
Class libraries (certified)	All classes	None
Clean-room development	Systems software	MIS software
Complexity analysis	All classes	None
Configuration control tools	All classes	None
Defect estimation tools	All classes	None
DoD standards	Military software	All others
Error-prone module analysis	All classes	None
Formal code inspections	All classes	None
Formal design inspections	All classes	None
Formal test plan inspections	All classes	None
Function point defect metrics	All classes	None
ISO 9000-9004 standards	Commercial software	Most other classes
Joint application design (JAD)	MIS software	System software
LOC metrics	None	All classes
OO analysis and design	Ambiguous results	Ambiguous results
Prototyping	All classes	None
Reverse engineering	COBOL primarily	Many languages
Quality circles	Ambiguous results	Ambiguous results
Quality function deployment (QFD)	Most classes to date	Still experimental
Quality measurement tools	All classes	None
Reusable code (certified)	All classes	None
Reusable designs (certified)	All classes	None
Reusable test cases (certified)	All classes	None
Reusable test plans (certified)	All classes	None

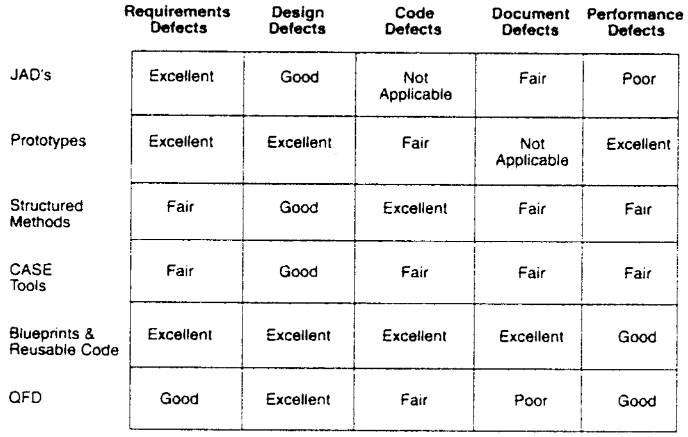
(continued)

Table 1 *(continued)*

Defect Prevention Approach	Software Classes With Good Results	Software Classes With Questionable Results
Risk analysis protocols	All classes	None
Test coverage tools	All classes	None
Test library tools	All classes	None
Total quality management (TQM)	Systems software	MIS software
Zero-defect programs	Systems software	MIS software

Defect prevention is a more difficult concept to grasp, and a more difficult concept to measure, than defect removal. This should not be a surprise, since preventive medicine is also more difficult to justify than curative medicine. In both domains, there is a strong synergy between prevention and cures and both are necessary.

It is useful to construct a simple matrix that shows defect origins on one axis and defect prevention effectiveness on the other axis. It is obvious from such a graph that there is no "silver bullet" that will magically eliminate all sources of error.



	Requirements Defects	Design Defects	Code Defects	Document Defects	Performance Defects
JAD's	Excellent	Good	Not Applicable	Fair	Poor
Prototypes	Excellent	Excellent	Fair	Not Applicable	Excellent
Structured Methods	Fair	Good	Excellent	Fair	Fair
CASE Tools	Fair	Good	Fair	Fair	Fair
Blueprints & Reusable Code	Excellent	Excellent	Excellent	Excellent	Good
QFD	Good	Excellent	Fair	Poor	Good

Figure 1: Software Defect Origins and Defect Prevention Effectiveness

Graphs such as this are useful for clarifying the kinds of defect prevention and defect removal methodologies that need to be utilized to achieve satisfactory quality levels. It is obvious that each source of error should utilize prevention methods that are "good" or "excellent" in order to minimize down-stream problems.