



QUALITY ASSURANCE IN A RAPIDLY CHANGING WORLD

V. Bradley and M. Kimmich, Eds

Brookes Publishing, Baltimore, MD

2003

**Chapter 14: SELECTING APPLICATION SOFTWARE / SERVICES TO IMPROVE AND
ASSURE SERVICE QUALITY: IMPORTANT CONSIDERATIONS**

John Ashbaugh

PURPOSE AND SCOPE

Agencies serving persons with developmental disabilities need to become “smarter” in their use of information technology. Agencies are struggling to maintain and improve the quality of their services in the face of the ever-growing demand for services and supports, a shrinking labor pool, thinning resources and the increased dispersion of consumers and supports in the community. Currently available information technology can help enormously. The electronic exchange of messages via email, electronic sharing of client records and electronic scheduling can contribute to more efficient coordination of services and supports by staff, contractors, consultants, volunteers, and others involved in the support of persons with developmental disabilities.

In addition to the benefits for agencies, increased reliance on information technology can help direct service workers and consumers. Direct service workers must gain relief from the burgeoning paperwork now attending the delivery of services and supports. Workers now spend significant amounts of time entering the same information on countless forms; such time would be better spent in service. Likewise, quality review teams spend precious time looking through case records, when such work could be done remotely through electronic “desk audits”—electronic queries allowing reviewers read-only access to client records.

Using electronic means, consumers and families can access needed information on the appropriateness and quality of services and support offerings, enabling them to make more informed choices about service providers, and to be more active participants in the service and supports process.

Demand for performance information is steadily growing at the national and state levels.ⁱ Agencies must respond without overly intruding into people’s homes and day arrangements, and without overburdening direct service workers. Workers can capture the needed data electronically in the normal course of their work, through the use of electronic input devices and electronic records systems. The time and cost involved in converting these data into useful form, i.e. into information for decision-makers, can be greatly reduced through the use of today’s powerful relational database, spreadsheet, statistical and report-generation software. The time and expense for decision-makers to access this information can be greatly reduced through the far-reaching, worldwide web—the internetⁱⁱ.

An abundance of application software and services is now available commercially that can be “applied” to meet these and other needs of end users. In this chapter, application software is variously referred to as “software package(s)” in the case of desktop and client/server software, “software services” in the case of web-based software, and “software,” “packages,” or “services” generally. The in-house development of software solutions (application software) makes little sense given the wealth of packaged software that can be configured or built-out to meet the “special” needs of developmental disabilities agencies today. The difficulty is deciding among the dizzying array of packages and services.

This chapter presents valuable information for every agency considering an investment in information technology for quality improvement and administrative purposes. It speaks primarily to agencies delivering direct supports, because this is where the demand for information management is the greatest. This chapter focuses on computer application software and services designed to improve the delivery of supports to persons with developmental disabilities. It defines and discusses several important considerations that an agency must weigh as it decides what application software will work best.

SOFTWARE SELECTION CONSIDERATIONS

Functionality

What functions does the software perform, how well, and what is the value of these functions to the agency and its clients?

Assuming that the vendor can demonstrate that the software indeed performs the function as expected, what is the practical value of the function to the agency? Practical functions might include such things as mailing list management, scheduling, management of facility and vehicle maintenance, fund raising, individual planning/budgeting, billing, work flow management, reporting, and the automatic generation of form letters or reminders triggered by dates or events.

Agency decision-makers should be realistic. Which of the functions will the agency really use? Most software functionality goes unused and thus carries no value. Sometimes agencies never intend to use these functions; they are known to be extra baggage at the time of purchase. At other times agency users are unaware of the functions. Too often the agency expects to make use of more functions than it actually does, perhaps because the benefits that come from using the function are not worth the cost of setting it up and learning to use it. It is best to ignore “nice to have” functions, recognizing that these are not likely to ever be used.

The quick way to decide among a variety of software packages in terms of function is to list the functions advertised, and to scratch out those of little or no use immediately or in the foreseeable future. Which package has the most functions of real value to the agency?

First and foremost, choose software that offers the functions needed by the agency.

Single-Function Versus Enterprise Applications

Application software typically performs one basic function such as fund-raising, billing, human resources, case management, payroll, or accounting. It consists of code written to perform these data-based functions and a database structure to hold the data. Application software designed to include accounting and most or all other functions involved in running an organization is commonly termed “enterprise” software. The enterprise software typically has a number of modules (basic functions), each of which correspond to a single-function software package or service. However, in the case of enterprise software, the code and database are unified.

In the experience of this author, the single-function applications tend to have more user-friendly features (e.g. ease with which queries can be made) than the enterprise packages; they have usually been around longer and thus are more highly evolved than their enterprise counterparts. Those favoring the use of a combination of single-function software packages, as opposed to an enterprise package, often use the term “best of breed” software to describe these packages, alluding to their functional superiority.

The advantage of enterprise packages is the unified (integrated) code and database. Users benefit in having to learn and use only one package and deal with only one vendor. Time is saved 1) by not having to enter the same data multiple times into separate (non-integrated) applications, and 2) by not having to enter data generated in one application into another to generate a report or bill. When databases are not integrated, and since it is fairly time-

consuming to reenter data, it may be necessary to pay a programmer to build electronic bridge(s) to port (transmit) data between applications; such costs are avoided by using enterprise applications.

Customers should be cautious: not all individual packages compare favorably to enterprise packages in terms of functionality and user-friendliness, nor are all enterprise packages fully integrated. Repeated data entry and manual data transfers into different modules of the enterprise package are sometimes necessary, albeit to a lesser degree than would be required when using non-integrated applications.

If one "enterprise" package offers the same functionality needed by an agency as a number of single function packages and if these packages are otherwise comparable, the one enterprise package is the better choice.

Flexibility

Raw functionality is not enough. The software must be flexible enough to meet the peculiar and changing needs of the agency and work in combination with any existing agency software (legacy software) worth saving. The flexibility of an application refers to the relative ease with which users are able to make changes to the application database, screens, queries and reports. Applications range from highly inflexible to highly flexible. At one end of the spectrum are packages where any changes to the computer screens, database, queries and reports must be done by the vendor (or trained programmer). At the other end are packages where changes to the computer screens, database, queries and most reports can be done by the user. Agencies will inevitably need to make changes to the data, screens, queries and reports to meet their changing needs; more flexible application software will allow them to do this in less time and for less money.

The data fields, screens and reports in less flexible packages are largely pre-defined. Where agencies need data, screens and reports beyond the standard, they must purchase programming time to have them customized. Customization can easily push the cost of systems beyond the reach of most agencies. Moreover, as it is the vendor's job to support users of their applications, they are reluctant to allow too much customization given the difficulty of supporting the widely varying system set-ups that result.

The most flexible software is object-oriented.ⁱⁱⁱ The functionality, screens and databases of object-oriented software are much easier to change than the functionality, screens and databases of non-object oriented software. Object-oriented software is also less prone to crashing when substantial changes must be made to the code, and is easier to integrate with other applications. The market is now insisting on object technology; all of the major vendors of the relational database management systems (RDBMS) prevalent today are working on object-oriented makeovers of their systems

Some object-oriented software is designed to fit the needs of a variety of industries and agencies, e.g. Intersystems' Cache. Other software is tailored for particular industries, (e.g. Danic Tools is designed specifically for health and human service agencies). Where available, industry-specific software is better, as it will be easier and less costly to configure and implement than the generic, cross-industry software.

Choose object-oriented software over conventional software of comparable functionality.

Desktop Versus Client/Server Software

The next consideration in selecting information technology is whether the agency's individual computers need to be linked together for the purpose of sharing data and/or programs (code). They might be networked using dedicated lines, or they might connect via modem over phone lines or via an internet service provider (ISP) over the internet. Desktop software is designed for use on a single computer, on which resides all software and data that the computer user needs. Client / Server software is designed to work on a network of computers, where a dedicated computer (called a "server") handles some of the processing tasks, while multiple smaller computers (called "clients") complete other processes by tapping into the server's central database and programs.

A rich array of desktop software is available off-the-shelf. The widest selling is Microsoft Office Suite which includes Word (word processing), Excel (spreadsheet analyses), Outlook (messaging), Powerpoint (presentation graphics), and, in the Professional version, Access (relational database management). Other popular Microsoft desktop software might include Front Page (web-site building), Project (project management), Visio (diagramming), and Publisher (desktop publishing).

In all cases, "Desktop Software" requires each desktop computer to have a licensed copy of the software on it. The software itself is not shared. While files generated by "Desktop Software" might be stored on another computer, a "file Server," and shared by multiple desktop computers across a network, the number of users who can efficiently make changes to it simultaneously are limited. This is because while the individual desktop computer can access a file, it is the desktop computer that must do the related computing, not the "file server". The desktop computer "borrows a copy" of the file from the file server, and, after using it, sends (saves) the changes back to the file server, overwriting the original file. For these types of applications, the file server simply acts as a central place to store and access files. It does not itself do any of the related computing. As databases increase in complexity and size, and as the number of users increases, the amount of data that must be transmitted from server to desktop computer and back quickly becomes too much for the desktop computer and/or the network to manage. If the network is accessing the file across phone lines or the internet, transmission times become unacceptably slow. This is not the case with "Client-Server" software.

Client / server software is designed to minimize the volume of information that must travel from computer to computer, be it over a network, phone line, or across the internet. While the software is still housed on each user computer (client), the database is housed centrally on the application server. Clients connected to the server all of the time (24 hours, 7 days a week) register changes to the database as they occur. Clients that connect to the server from time to time and use their own version of the database in the interim, synchronize their "remote" database with the central database by exchanging only the changes made to the central and client databases during the interim period. In either case, the transmission of entire databases from computer to computer is avoided. If a client/server database contains a mailing address that changes, only the changed address needs to be transmitted from the source computer to the other computers sharing the application, not the entire database as would be the case with desktop software. Likewise, processing tasks that would generate a high volume of network traffic were they conducted on a central computer (the server) and distributed to other computers on the network, can be programmed for execution on user desktops to lower network traffic—for example, client/server software might be configured so that each of the computers on the network (clients) are able to generate their own reports using data from the central database residing on the server, in order to avoid the much higher volume of network traffic that

would come from the centralized generation and distribution of reports to the networked computers.

Other factors favor consideration of client /server software. Client / server databases are becoming increasingly sophisticated, easier to set up and maintain than desktop application databases. The most popular systems for very large organizations are Oracle and Informix; for mid-size and smaller organizations—the vast majority of DD agencies--Microsoft's SQL Server. Not long ago, setting up an Oracle or SQL database would take a significant amount of time because the system administrator would have to individually set and test more than fifty parameter settings, adjusting as necessary to optimize performance. Today, a client / server database can be set up in a day or two , with the system setting the optimal parameters automatically and dynamically.

If more than approximately a dozen users are or will be sharing an application and database of any size and complexity, agencies should give serious consideration to client/server as opposed to desktop applications, to avoid deterioration of performance.

Client/Server Versus Web-Based Software

Despite its advantages over desktop software, client/server software has inherent limitations, most notably, the challenges involved in keeping the software running on high numbers of widely dispersed computers (clients). As agency systems grow, the work and expense involved in troubleshooting system/user problems and installing software upgrades on these geographically-dispersed machines (clients) can severely tax an agency's information technology (IT) staff or consultants. The job of system support becomes even more difficult when agencies find themselves with a range of unconnected, even incompatible, makes of hardware and software.

Web-based software uses a standard "open" language, allowing computers of different makes and models to connect with one another and to share text, graphics, images, sounds, and video information over the internet. Unlike client / server applications, both software and data reside and typically operate on a central server and can be accessed using internet browsers such as Netscape or Microsoft Internet Explorer--standard on today's PCs.

Web-based applications are characteristically delivered and supported over the internet by application service providers (ASPs). ASP offerings typically include the installation, operation, upgrade, and support of application software; the software and data reside on the ASP server(s). User agencies make periodic payments to the ASP for services rendered over a multi-year contract period.

At this time, a number of well-established ASPs offer function-specific applications (e.g. billing, accounting, payroll) across a number of industries. Still others offer Microsoft and other widely used off-the-shelf applications to customers (USInternetworking, Interliant, FutureLink) However, most ASPs provide applications on an industry-by-industry basis, e.g., insurance, advertising, financial services, and health care. WebMD, formerly Healtheon, is the highest profile ASP in the health care field.

Web-based systems are the wave of the future. Their ability to connect disparate application and operating software and hardware will not only facilitate the development of agency-wide enterprise systems linking long-separate clinical and back-office applications, but, in the near future, will foster the development of extended multi-enterprise systems, that is, systems linking service agencies, consumers, advocates, funders and others in the DD

community. These systems can run as a virtual private network (VPN) over the internet with encryption and firewalls^{iv} providing the necessary security. The user-interface with internet-based applications is simple and the training, simplified.

Because the web-based software and database can reside and execute wholly on the server, agencies can avoid the headaches and costs associated with the installation, upgrading and support of software resident on widely-distributed PCs (clients).. For the same reason, many personal computer (PC) malfunctions caused by erring users are avoided.

ASP arrangements can yield significant benefit and savings for agencies. Agencies avoid the hassles and cost of themselves recruiting and retaining technical staff to maintain and troubleshoot the system and to keep the system database, reports, and actions in tune with changing agency demands. They avoid the risk associated with maintaining and owning malfunctioning, obsolete hardware and software. The ASP offers application-specific expertise and enough depth to mitigate the effects of staff turnover. Particularly important to most DD providers, ASPs provide the means by which subscribing agencies can share many of the costs associated with operating information systems.

To date, most web-based services have not yet developed some of the functionality and user-friendly features of their more seasoned client / server counterparts. However, the functionality and features of web-based systems are growing rapidly. Out of necessity, client/server software developers are web-enabling their systems by building their own web-enabled user interfaces or by using terminal emulation software. Terminal emulation software enables client/server applications to reside and execute wholly on a server with browser access, i.e. to operate as web-based software. Most terminal emulation software is licensed by vendors such as Microsoft, Lotus, and Oracle and other vendors to work with their own software. Citrix^v terminal emulation software is designed to work with the software of all different vendors, but users must pay a premium for its use.

The benefits of web-based services have yet to be fully realized. Whether the application software is web-based or web-enabled, the benefits derived from the ability of these systems to execute wholly on the server are limited by the narrow bandwidth (limited capacity) of the lines by which most agency users can connect to them (over the internet). As a rule, the more robust the application, the greater the volume of information that must be transmitted from server to client, and the wider the bandwidth must be to accommodate.

Consequently, developers of more robust web-based and web-enabled systems are finding it necessary to house a portion of their software or adjunct software (e.g. report generation) on user PCs, to avoid the transmission of large volumes of data from server to user and thereby maintain an acceptable level of performance over the low-bandwidth lines prevalent today. In so doing, they effectively compromise the most important advantage that web-based or web-enabled software holds over client/server software— centralized, non-distributed software.

The agency can fairly easily determine the effect of such a compromise. The vendor will require the agency to install the software or a portion thereof on user PCs. To assess whether the software will perform at an acceptable speed over the bandwidth available to the agency, the agency should actually test the software as it will be applied at the agency; if this can't be done, the agency should check the software's performance with other agencies having like requirements; and if this can't be done, the agency should condition payment on the achievement of promised performance (speed).

Choose web-based or web-enabled software if it has the functionality and flexibility desired, and if the application has shown it will perform at an acceptable speed, and if the advantages of web-based over client/server and desktop packages are material to the agency.

System Costs

Many costs are associated with the procurement, implementation and operation of application software, some immediate and obvious, some less so. It is important that all are considered. The bottom-line question is: What is the total cost associated with the ownership or use of an application?

The fundamental choice for agencies today is between client/server and web-based applications, the latter most always provided by a vendor or other application service provider. Agency decision-makers need to understand the differences in the costs of these two principal modes of application procurement and deployment; they must be sure to recognize all of the costs associated with these two alternatives..

Software Costs

Client/server From time-to-time upgrades of the application software will have to be purchased and installed. Software, in addition to the application software, is required. Client/server relational database software is required to manage the network connections and requests from multiple users. Software utilities are needed for the back-up of the system and to set up the firewalls necessary to prevent unauthorized access to the system.

Web-based All software and upgrades required for the operation of the application would be provided by the ASP. The firewall can often be purchased and maintained by the Internet Service Provider (ISP)^{vi} through which the agency connects to the internet. While the ASP and ISP would pay the same amount to purchase the software as the agency does under a client/server arrangement, the ASP and ISP can spread these costs among the many agencies subscribing to their service, thereby reducing the cost to each agency.

Hardware Costs

Client/server A server dedicated to the particular application(s) is required. User PCs may have to be upgraded or new ones purchased to obtain the random access memory (RAM), processor speed and/or the off-line storage needed by the application.

Web-based Just as with the client/server application, an application server is required. Lower capacity, less powerful PCs may be required by the agency, but this will not necessarily be the case; as indicated earlier, the web-based software may be designed to run in part on user PCs.

Network Costs

Client/Server Network bandwidth (speed) may have to be increased to manage the added workload associated with the application. Additional or upgraded switches and routers may be needed to manage the added traffic on the network associated with the application. The operating system (e.g. Unix, Linux, Windows, NT) may have to be changed to match the operating system on which the application is designed to run, to avoid the extra time that would otherwise be required for middleware to crosswalk between the application and foreign

operating system. Alternately, the bandwidth on the network lines could be increased to provide the extra speed necessary to compensate for the added translation requirements.

Web-based In addition to any costs for the build-up of the agency network, the agency must bear the cost of the internet connection, and the costs of the primary and backup lines to the off-site application server run by the ASP.

Implementation Costs

Client/server Implementation involves system configuration, data migration, bridge and report building, installation and testing, training and support. Nearly every system needs to be configured to some extent to best meet the needs of the agency. System configuration may involve screen design, database structuring, the setting of rules to automatically trigger decisions and actions based on dates and events, and other tinkering necessary for the application to work within the agency infrastructure. The work involved in configuring the system may be limited in the case of relatively inflexible software, but can be high in the case of highly flexible (configurable) software. Software typically comes with an array of standard queries and reports. Other queries and reports must be custom built.

If the application is to accept data from, or generate data for other applications, bridges may have to be built (code written) to port these data. The one-time porting of data from existing databases to new ones may be managed using the import / export utilities in the software applications involved. However, where a significant amount of data must be ported from one application to another on a continual basis, a bridge is generally required.

Web-based While the same types of implementation tasks are entailed, the amount and cost of coding required in the configuration and customization tasks will generally be greater with the web-based systems. This is because the web-based systems lack the flexibility and functionality of their client/server counterparts and are thus likely to require more “work-arounds” to meet an agency’s requirements, and because web-based coding is generally more expensive than the coding of non-web-based software, given the higher power of the client/server languages and higher compensation demanded by web-based coders.

Installation and Testing Costs

Client/server The software must be installed on the server(s), PCs and other hardware, and then tested. In addition to the initial testing by the vendor over a period of days, often a longer trial period is needed, over a period of weeks or months, during which the users test the system and their ability to manage it before the installation is considered final. Users must be trained. While all vendors provide initial training, it is generally most cost effective for the agencies to provide their own subsequent training.

Web-based The software must be installed on the ASP server. Some software may also need to be installed on agency PCs. The bulk of the installation and testing is done by and within the ASP. User training costs can be minimized through the use of remote (distance) training arrangements.

Support Costs

Client/server Users invariably require vendor support; the amount required should decline over time. Support should include the cost of software updates (minor changes in code

and functionality) and may include the cost of software upgrades (major changes in code and functionality). In the case of applications with databases of any size and complexity, an agency must dedicate a full or part-time position to the implementation and subsequent support of the application. Responsibilities typically include: 1) database modification and development, 2) data quality control, 3) system querying and reporting (beyond the standard queries and reports), 4) compilation and maintenance of a system policies and procedures manual governing system use, and 5) working with the network administrator (staff or consultant) to assure a functioning agency network.

Web-based The amount of time required to manage the five responsibilities noted above is comparable for client/server and web-based systems. However, an inherent difference with web-based systems is the amount of time and related costs required to 1) install software updates, 2) train users, 3) trouble-shoot the system, and 4) provide system security. Installing software updates and upgrades on agency computers is unnecessary if the web-based software resides wholly on the ASP server. The amount of time that must be spent in the training of users should be less to the extent that the web-based software comes with internet-based training programs that can be used to facilitate or substitute for scheduled in-house agency training. The amount of user support should be less, given that the opportunity for user error and PC malfunction, the source of many problems, is minimal in cases where web-based software and database reside on the central ASP server, not on user PCs.

The costs of providing system security should be substantially lower with web-based systems delivered by ASPs. Most DD agency systems contain electronic records pertaining to the support of persons funded by Medicaid, Medicare or private insurance, and thus the security requirements associated with the Health Insurance Portability and Accountability Act (HIPAA)^{vii} apply. The HIPAA requirements are prohibitive. One of the most significant differences in an agency's costs of supporting client/server and ASP applications will be in complying with the HIPAA requirements. HIPAA guidelines strongly suggest that the responsibility for maintaining a secure environment for an agency's electronic data be assigned to a security officer or organization, and that the agency appoint a privacy officer responsible for assuring client confidentiality. Even though, in smaller organizations, one person could fill both positions, the workload and expense implied is prohibitive for all but the largest agencies. The costs of site security will be far less for an ASP serving many agencies from a single site than for an individual agency with data residing on servers and PCs across many sites.

Debate continues on whether the authentication (verification of the identity) of users can continue to be done using passwords over virtual private networks (VPN) -- common practice today — or whether the more sophisticated public key infrastructure (PKI) system (akin to the smart cards used by automated teller machines) should become the standard. The cost of implementing the PKI method is significantly higher. If it becomes the recognized standard, the cost to individual agencies will be prohibitive. This will create another economic advantage for ASPs where the cost can be shared among subscribing agencies.

Cost Sharing

Many of the costs associated with acquiring and deploying an application are about the same whether incurred by an ASP or individual agency. However, because the ASP is able to spread the costs among a number of agencies, the cost to subscribing agencies is a fraction of what the cost would be to an individual agency. The costs and risks associated with the acquisition, implementation and operation of client/server application software are too much for smaller agencies to manage.

As a general rule, smaller agencies (under \$5 million in annual revenue) would be well-advised to look for ASPs offering the applications they need, rather than trying to acquire and maintain client/server software themselves.

SUMMARY

Application software can be used to considerable advantage by agencies serving persons with developmental disabilities, allowing them to better manage the delivery of services and supports and assure their quality. However, the time and expense involved in procuring, implementing and maintaining these applications is generally significant. Agencies should apply due diligence as they proceed: Look first for proven functionality of real value to the agency. Choose the fewest packages (i.e. different vendors) to get it. Make sure the software is flexible enough to meet the agency's particular and changing needs over time, and to work with any existing applications worth saving. Consider carefully what combination of desktop, client/server or web-based systems are the best fit for the agency, given the resources at hand.

Applications available to agencies serving persons with developmental disabilities continue to improve in functionality, flexibility and affordability. Some agencies hesitate to make a substantial investment in application software or services today, for fear that better packages will appear tomorrow. Indeed, better packages will appear. The real question is whether the investment in the application(s) being considered is justified today by the promised improvements in the agency's operation.

Some agencies continue to look to the state DD or Medicaid Authorities to develop the systems needed. However, there is no reason to believe that the shortcomings of the state mainframe systems in years past will not be repeated with the client/server and web-based systems of the future. State systems will continue to be long on the functionality needed by the state, and short on that needed by the front-line agencies and workers. States will invariably run short of the funds and resources needed to properly maintain and support its systems.

While this chapter has addressed some of the most important considerations in choosing software applications or services, it has not begun to address the full range of factors to be considered in selecting application software or services. Weighing many of these considerations demands an understanding of information technology that comes only through years of training and experience. If agency staff do not have the technical knowledge and experience required to judge the application's workability, look to Board members, outside volunteers or paid consultants. Judicious selection and use of application software can make a critical difference in DD agencies' ability to assure and improve service quality.

ⁱ A handful of efforts are already underway at the national level to obtain indicators on the performance of DD systems: Quality Outcomes Project (ACC), Core Indicators Project (NASDDDS), and two MR/DD-related Performance Indicator projects in the Centers for Medicare and Medicaid Services (CMS, formerly HCFA), with dozens more at the state level.

ⁱⁱ The Internet is an international network of computers that operates on a **backbone** system without a true central **host** computer. Today's Internet links thousands of universities, government institutions and companies, but when it was created in the 1960s, the Internet linked just four computers. Technically, the Internet and the **World Wide Web** are not interchangeable terms; the Web is an integral child of the Internet whose ease of use has made it much more popular than its less graphical parent.

ⁱⁱⁱ Object-oriented software is made-up of objects. An object is a software module that contains a collection of related procedures and data. Each object has a basic purpose or function it performs (written as code). Unlike non-object oriented software where code and data are separate, each object contains the data used in performing the

function as well as the code. It is self-contained. Users make use of each object through a defined set of messages, and can link the objects to one another to perform larger and broader functions.

^{iv} A firewall is a security device situated between a private network and outside networks. The firewall screens user names and all information that attempts to enter or leave the private network, allowing or denying access or exchange based on pre-set access rules.

^v A stand-alone Citrix server is required in addition to any servers required for the application software.

^{vi} An Internet Service Provider (ISP) is a company that provides modem or network users with access to the **Internet** and the **World Wide Web**. Although some ISPs charge by the hour, most offer monthly or yearly flat rates. Recently, telephone companies have begun to address the notion of combining Internet access rates with local telephone service.

^{vii} Enacted by Congress in 1996, this Act, among other things, requires organizations that receive Medicaid, Medicare and Private Health Insurance and that maintain or transmit client information electronically to implement security measures to guard the integrity and confidentiality of client information.