

A Guide to Evaluating Data Entry Systems

A White Paper



VIKING SOFTWARE SOLUTIONS

A Division of Phoenix Software International

**6660 S Sheridan Rd, Suite 202
Tulsa OK 74133-1730
(918) 491-6144
Fax (918) 494-2701**

Table of Contents

Introduction.....	1
Eight Steps to Evaluating Data Entry Systems.....	1
Step 1: Functional Requirements List	1
Step 2: Compare and Rank the Software Alternatives.....	1
Compare to Functional Requirements	1
Technical Support.....	2
Vendor's Reputation	2
Vendor Commitment to their Product	2
Step 3: Cost/Benefit Analysis	2
Data Entry Labor Costs	3
Step 4: Select the Software Package.....	4
Step 5: Configure the Hardware for the Selected Software	4
Step 6: Order the Hardware and Software	4
Step 7: Develop the Implementation Plan.....	4
Step 8: Implementation.....	4
Installation.....	4
Training.....	5
Production Cut-Over	5
Post-production review	5
Appendix A: Features and Functions Checklist	6
Data Field Attributes	6
Edit Validations.....	6
Data Entry Screen Features.....	7
Job Creation and Screen Generator	7
Data Entry System Features	7
Appendix B: Other Considerations.....	9
Data Accuracy	9
Early Error Detection.....	9
Cost of Data Errors	9
Automated Data Entry Systems.....	9
Double-Key Verify.....	9
Error Correction Costs	9
Cost/Benefit of Double Key Verify	10
Operator Performance	10
Measuring Performance	10
GUI Interfaces	10
Application Development Tools.....	10
Conservation of Scarce Resources	11
Multi-Platform Capability.....	11
About the Company	11

A Guide to Evaluating Data Entry Systems

Introduction

The life cycle of data entry systems is often more than ten years. Because of this long life, only a few people have the opportunity to evaluate more than one system in their career. Many evaluators, who are doing it for the first time have asked Viking for assistance in organizing the evaluation plan. This guide is intended to bring the benefit of our staff's many years of experience to those who are faced with this task for the first time. For those who are experienced in this activity, we hope that they will find some useful ideas and checklists. Perhaps even a new wrinkle or two.

Evaluating data entry systems can be as simple or as complex as needed. It is important to do a careful and thorough evaluation within reasonable time and cost boundaries based on the size and value of the project. Failure to do the evaluation properly results in incorrect decisions. Over zealous evaluation teams can spend more money considering the project than the project will cost. Either extreme is equally bad.

Following are the steps to evaluate data entry systems. Each of these steps will be addressed in detail.

Eight Steps to Evaluating Data Entry Systems

1. Develop a list of functional requirements for the new system.
2. Compare and rank the software alternatives.
3. Perform a cost/benefit analysis of the top software candidates.
4. Select the software package.
5. Configure the hardware for the selected software.
6. Order the hardware and software.
7. Develop the implementation plan.
8. Install, train and implement.

Step 1: Functional Requirements List

The functional capabilities of the data entry system are first defined in rough general terms. Begin by listing the most important characteristics required for your applications. Also, identify the deficiencies of the existing system. Continue to refine the list to help in the evaluation and comparison of various systems.

It is important for all parties in the decision process to come to an agreement of which basic functional capabilities are required and what are the overall objectives of the new data entry system. A key activity is to decide how many people will be using the data entry system and if they will be full time users, or if data entry will be only one of their tasks.

Appendix A contains a list of data entry system attributes. Use it to help build the requirements list and to set priorities for its items.

Step 2: Compare and Rank the Software Alternatives

Reviewing and comparing the software is often the most difficult and time-consuming portion of the evaluation. However, it is the most important part because of its affect on labor costs through its influence on operator productivity. The data entry labor costs are by far the largest cost component of the entire system.

Compare to Functional Requirements

It is important to review the candidate packages to see if they meet the functional requirements defined in Step 1. Ease of use, operator performance, service and reliability are the key factors to evaluate.

Technical Support

It is easy to compare the technical support from the vendors. As the system is being tested or reviewed, call the technical support hotline and ask questions. How long does it take to get answers? Are they reasonable and correct? Do the support people seem competent and do they understand data entry? Are they friendly and concerned about your problems and questions? Do you sense a rapport with the technical service reps? Be wary if you do not get satisfactory results when you are just an evaluator. Service will be no better, and probably worse, after you have paid your money and become a customer. Remember, you will be dealing with these people long after the initial evaluation period.

Technical support should go beyond the simple "how do I do it" to include "what is the best way" and "why is this important." The vendor should recognize the importance of caring for their customers by helping them develop successful and productive data entry applications.

Vendor's Reputation

Ask the vendor for a list of people using their product. Call the users and ask what they think of the product and the company. Ask about their experience with technical support.

Vendor Commitment to their Product

The vendor's commitment to their product is also easy to check. Ask what version they are delivering today? When was it released? When were the last four or five versions released? Request a copy of the product release notes to see what was in the last two versions. A company committed to its product and its customers will release a new version with significant enhancements every year or so. Long intervals between releases and no major enhancements are a sign that the vendor no longer considers it an essential part of its future. Try to find out if they are simply milking it for the last few sales before eliminating active support of the product.

Step 3: Cost/Benefit Analysis

The cost/benefit analysis of the top contenders involves the following activities.

- A. Identify the costs of the new system.
 1. Software costs
 - a. Perpetual license fees
 - b. Click based fees
 - c. Leased access fees
 - d. Maintenance costs
 2. Hardware costs
 - a. Purchase or lease costs
 - b. Maintenance costs
 - c. Intranet
 - d. Server
 - e. ASP
 - f. Remote access
 3. Conversion costs
 - a. Internal staff costs
 - b. Outsourcing costs
 4. Training costs
 5. Ongoing data entry labor costs
- B. Identify the benefits of the new system
 1. Tangible benefits

- a. Cost reductions
 - b. Savings
 - 1) Labor costs
 - 2) Computing charges (e.g., mainframe)
 - 3) Hardware maintenance cost
 - 4) Leased equipment
 - 5) Software maintenance
 - 6) Supplies and consumables
 - c. Process improvements
 - 1) Reduce errors and corrections
 - 2) Enter information ONLY once
 - 3) Streamline information flow
 - 4) Put editing at the point of entry
2. Intangible benefits
- a. Data accuracy
 - b. Staff morale: existing, future
 - c. Ability to meet expectations of clients
 - d. Productivity
 - e. Environmental
 - 1) Noise
 - 2) Ergonomic
 - f. Flexibility for future changes
 - 1) Hardware/server/intranet
 - 2) Software
 - 3) New & changed requirements
 - 4) New technology
 - 5) Different organizational structures
 - 6) Work load variations

Appendix B contains detailed discussion of some topics that influence the cost/benefit analysis and it also has examples of cost calculation methods.

Data Entry Labor Costs

To calculate the data entry labor cost savings in B.1.b. of the outline above, it is necessary to estimate the present costs and the expected costs for the different systems.

The average data entry system will have an expected life of five to ten years. Direct and overhead costs of data entry operators will be \$100,000 to \$200,000 per operator. This is twenty to fifty times the original cost of buying, installing and maintaining hardware and software over the lifetime of the data entry system.

Some 90-95% of the life-cycle cost of a data entry system is labor. This is why a proportionate consideration must be given to the operator productivity factors that so greatly influence the total costs.

Error correction is another important factor in the labor cost of operating a data entry system. Examples of how to calculate the cost of errors are in Appendix B.

Step 4: Select the Software Package

Once the cost/benefit analysis has been completed for each contender the selection process begins. When there is no clear-cut economic difference between the alternatives, the decision will be made based on subjective factors such as the vendor's reputation, technical support, etc.

Step 5: Configure the Hardware for the Selected Software

Configure the hardware after selecting the software package. Most of the information needed for this process was gathered in Step 3, the Cost\Benefit Analysis. Your organization may have experts to help configure the computer hardware.

Step 6: Order the Hardware and Software

Place the orders with the hardware and software vendors. Your organization probably has a purchasing department or other people to help with procurement.

It is important to select a reliable hardware vendor who can respond quickly with the proper personnel and parts. The services required from the hardware vendor will depend on the availability of expertise in your organization. Look for hardware vendors who can provide the necessary installation and maintenance services.

Pay specific attention to the delivery dates of the hardware and software. They are an important influence in establishing schedules for the implementation plan.

Step 7: Develop the Implementation Plan

Once the hardware and software have been ordered, it is time to create the implementation plan. This plan involves the following topics.

1. Space renovation and preparation.
2. Software training for developers and supervisors.
3. Hardware usage training.
4. Converting existing data entry jobs.
5. Developing new data entry projects.
6. Data entry operator training.
7. Revise and document affected systems and procedures.
8. Notify the affected client groups.

Step 8: Implementation

Implementation step consists of doing the tasks outlined in the plan developed in the previous step. Implementation involves four phases:

1. Installation
2. Training
3. Production Cut-Over
4. Post Production Review

Installation

Installation can be as simple as loading an application software package or as involved as the physical replacement of computers, terminals, printers, tape drives and renovating office space.

Training

Training is often a neglected aspect of a project. It should begin with communication from the highest levels of the organization to all affected personnel telling the reasons for the changes and stressing the positive impacts they will have on the organization. Change causes stress and anxiety and therefore people are naturally resistant to change. Bringing the staff into the process fosters ownership, understanding and cooperation and minimizes the "this is the way we have always done it" syndrome.

Follow the preparatory briefings with the necessary training to give personnel a proper understanding of the new system. The new data entry system will never be exactly like the old one and important decisions will be made regarding modifying software versus modifying procedures. A good understanding of the software is necessary to make well-informed decisions and the investment in training will be saved many times over during the next phase.

Production Cut-Over

Production cut-over is the most visible milestone of the project and can be heralded by much fanfare or it can pass quietly depending on the overall impact of the project. Often a period of parallel operation of the new and old systems is warranted to reduce unanticipated problems. Select the period of the parallel operation to balance the additional workload on the staff against the benefits and risks. Full value of the new system is achieved only when the people are eager to use it!

Post-production review

Place checkpoints in each of the preceding phases to monitor progress and allow personnel to proactively address issues as they arise. Documenting and publicizing the savings and improvements due to the new system re-enforces the sense of ownership. The project team members get a greater sense of accomplishment and favorable exposure to management.

Conduct periodic reviews to measure the benefits associated with the new system and to highlight areas for further improvement. New software releases provide excellent opportunities to take advantage of new features and improve the system. Continual improvement is an important goal.

Each of these phases are important elements in implementing the new system. Headaches result from neglecting any of them and may jeopardize the success of the project.

Appendix A: Features and Functions Checklist

- Multiple Computer Platforms
 - Windows
 - DOS (LAN, Stand Alone)
 - Linux
 - UNIX (most variants)
 - VMS
 - Other
- Scroll Fields
- Menu Based System
- Functions accessible through keyboard
- Controlled access (ID and Password required)
- 029 Keyboard Simulation

Data Field Attributes

- Multi-line word wrap fields
- Allowable characters
- Numerics
- Upper Case Alphabets
- Lower Case Alphabets
- Special Characters
- Combinations
- National alphabets
- User defined character sets
- Blank or zero fill
- Left or Right Justify
- Required to enter data
- ENTER required to execute field
- Verify this field
- Protected fields
- Must key all characters in a field
- Default data values
- Conditional entry
- Display attributes
 - foreground color
 - background color
 - intensity
 - reverse video
- Non-display fields
- Location of field in output record
- "Overpunched" negative signs
- Leading or trailing +/- signs
- Auto keyboard shift
- Multi-line fields with word wrapping

Edit Validations

- Table lookups (sequential files)
- Acceptable value lists
- Table lookup with replacement in same & other fields
- Indexed tables (Multiple Field)
- Date validations
- Compute values & store in other fields
- Auto Increment
- Ascendancy Check
- Next field to enter based on data in field
- Valid postal codes
- Skip certain fields based on data
- Check digits
 - validation
 - generation
- User-written field edits using standard programming languages such as "C"
- Context sensitive Help Messages: multi-levels
- Dup. data from any previous field
 - manually

- Next program level based on data in field
- Edit masks
- Compare to other fields
- Compute values based on one or more fields

- automatically
- Immediate re-key verification

Data Entry Screen Features

- Scrolling fields for repetitive groups of fields
- Multiple screens
- Linkage: one screen to the next
- Uses most terminals & monitors
- Interrupt & restart jobs

Job Creation and Screen Generator

- Full screen - "what you see is what you get"
- Automatic documentation
- Use & display all screen attributes
- Easy to draw lines & boxes
- Copy & modify screen definitions
- "Cut & paste" facility
- Menu driven
- Easy to use
- Extensive error checking
- On-line help
- 99 Screen (record/level) types

Data Entry System Features

- On-line help for each field
- Application generator
- No programming required
- Menu based options
- Multiple entry modes
- Begin keying a new batch of data
- Modify existing data batch
- Key verify existing data batch
- Conditional Verify
- Blank Field Verify
- Sample Verify
- Examine data w/o modifying
- Search for record number or contents of record
- Verify Mode Error Correction
- Character Correct
- Field Correct
- Record Correct
- Data File Manipulation
- System & user supplied error messages
- Data validation & editing
 - 1) At the character level
 - 2) At the field level
 - 3) At the record level
- Multi-record validations
- Transaction Journaling
- Balancing multi-field validations
 - 1) In a single record
 - 2) Batch totals
 - 3) Batch subtotals
 - 4) Conditional Balancing
- Operator Statistics (Time, records, keystrokes, errors, etc.)
- Flexible file (batch) naming conventions established by job
- Rearrange the location of the function keys
- Use different keyboards

- Insert & delete records
- Modify records
- Screen navigation
- Character forward or backward
- Field forward or backward
- Record forward or backward
- Insert & delete character
- Replace characters
- Advance to field with special "tab stop" indicator
- Immediate error messages
- Require operator to acknowledge (optional)
- Provisions for hearing impaired operators
- Reformat & restructure data files without programming
- Standard data files accessible by other programs, utilities, etc.
- Prepare data files to be imported into spreadsheets, data bases & word processors
- Supervisor facility to monitor & control the data batches
- Backup & restore without affecting operator performance
- Security: control of tasks available to specific users & operators
- 4-GL File Reformatting
- Customizable
- Field Edits
- Exit Routines

Appendix B: Other Considerations

This appendix contains some brief dissertations on a variety of topics that are appropriate to evaluating data entry systems. They are grouped into four primary categories; Data Accuracy, GUIs and Windows, Application Development Tools, and Multi-platform systems.

Data Accuracy

It is impossible to overemphasize the importance of correct data. Data errors are often the most costly aspect of data entry, costing the organization even more than the original data entry. Unfortunately, many of these costs are intangibles or are not easily measured. But they are very real!

Early Error Detection

When errors are detected early in the data entry process the repair cost is very little --- only the time to re-key the correct character or field. It is very costly to correct errors that are not detected until after the data has been processed. There may be an hour or more of clerical time involved in finding the document and entering the corrected transaction. The time lost in processing errors can also have a substantial cost. For example: delayed shipments, late invoices, customer dissatisfaction, and other tangible and intangible costs.

Cost of Data Errors

Failure to detect errors is even more costly. Erroneous part numbers result in shipping the wrong product. Incorrect customer numbers lead to all sorts of unpleasant and expensive problems. Invalid quantities and amounts also cost the company far more than the cost of the data input.

Automated Data Entry Systems

One of the biggest problems with automated data input systems are data errors. The data validation and error correction portion of an OCR/ICR system is an often overlooked factor. An inadequate system for repairing errors can make or break such a project. Some methods for correcting errors are not well designed for high volume projects.

Double-Key Verify

Double key verify is a time-proven method of increasing data accuracy. Character sieves and extensive field edits are required to detect errors at the earliest possible time. Multi-field validations and database lookups eliminate errors and keystrokes.

Error Correction Costs

The following example illustrates the cost of correcting a single character keying error.

Assumptions:

- The national average keying rate = 12,000 keystrokes per hour (kph).
- Data entry operator costs with full overhead burden = \$15 per hour or \$.25 per minute.
- The average time to read batch error report and locate erroneous document from the files = 20 minutes
- The time to reenter correction transaction detected off-line = 1 minute

Calculations:

The cost to correct single character error detected on-line is:

$$\text{Cost} = 1 \text{ keystroke} / 12,000 \text{ kph} \times \$15 \text{ per hour} = \$.00125$$

The cost to correct errors detected off-line is:

$$\text{Cost} = (20\text{minutes} + 1 \text{ minute}) \times \$0.25 \text{ per minute} = \$5.25$$

The ratio of the cost of on-line error detection and correction versus off-line error detection and correction is:

$$\text{Ratio} = \$5.25 / \$0.00125 = 4,200 \text{ to } 1$$

Cost/Benefit of Double Key Verify

Assumptions:

- 99.9% keying accuracy
- Double key verifying corrects 95% of original errors.

Calculations

Cost to verify 12,000 characters:

$$1 \text{ hour @ } \$15/\text{hour} = \$15.00$$

Number of errors in 12,000 characters:

12

Cost to correct 12 errors off-line:

$$12 \times \$5.25 = \$63.00$$

Double key verify benefit:

$$\text{\$48.00 per 12,000 characters}$$

Operator Performance

The national average for data entry operators is 12,000 keystrokes per hour. Really fast operators exceed 20,000. There is no reason that today's systems should ever slow the fastest keyer.

Measuring Performance

Keystrokes per hour are not the whole story, however. Use them with some caution. Advanced keystroke elimination methods can affect the keying rhythm and reduce keying speed, but they should result in greater net throughput. Measuring records per hour, or transactions per hour, may be a better metric. Do not forget to account for correcting keying errors!

GUI Interfaces

The conventional wisdom is that a Graphical User Interface (GUI), is the easiest and fastest approach to data entry. Methods and techniques that are best for programmers and knowledge workers are often counterproductive for high-speed, high-accuracy data entry.

The "intuitive" feel of a GUI approach is great for non-repetitive tasks because they are not performed often enough to achieve proficiency. However, when entering large amounts of similar data, it is essential that the operator's hands never leave the keyboard. A mouse slows down a good data entry operator. A GUI approach becomes costly in production time.

Application Development Tools

The time to initially set up, and later to modify, data entry applications is an important cost element. Viking uses an interactive Forms Painter to define jobs. Continual development and refinement since 1980 has proved this to be the most efficient method. Substantial data entry applications are routinely developed in an hour or so by experienced staff. New users create the first application surprisingly fast.

The ability to copy portions of jobs and to "cut and paste" is an important time saver. The ability to assign default values and attributes can further reduce the time to develop data entry jobs.

Conservation of Scarce Resources

Typically, higher-cost knowledge workers, who are scarce resources, do job setup. These talented people are needed for other tasks, so it is important to reduce data entry application development time. Contracting data entry conversion is common to conserve these valuable resources.

Multi-Platform Capability

Typically, organizations prefer one computer platform over another. Today, a network is the most popular choice. As rapidly as computer technology is changing, who is to presume to know the platform of choice five or ten years from now? Many data entry applications have migrated from keypunch machines to key-disk machines to mainframe terminals to PCs to servers to intranets. Keep your options open for the future. It is cheap insurance!

Look for data entry software that is available on many platforms.

About the Company

Viking Software Solutions is a division of Phoenix Software International, Inc. with offices at 6804 South Canton Avenue, Suite 900, Tulsa, Oklahoma 74136.

The company was founded in 1980 as Viking Software Services. In 2000, Phoenix Software International, Inc. and acquired the company and renamed it Viking Software Solutions.

Viking specializing in Data Entry Application Software and excels in mission critical, enterprise level applications requiring Power/Precision Data Entry. The company's software products are:

- Viking Data Entry™ (VDE)®
- VDE+ Images®
- ImagEntry®
- Viking Scanner Module (VSM)™

The company also provides professional services including consulting, training, custom programming, systems re-engineering, cooperative and turnkey conversion services.