



# **BUILDING AN EFORMS STRATEGY**

---

**INFORMATIONAL ARTICLE**

## Building an eForms Strategy: Elements of a Successful Program

### Introduction:

A recent question posed to an industry newsletter wondered, "In the 1990s, many industry pros predicted rapid growth for electronic forms. Why aren't more clients using them, and are eForms still a viable growth area?" We believe that this question gets to the heart of the matter and its answer, while complex, points to the need for an effective enterprise strategy for electronic forms.

There are many issues that impact the answer to this question. They include a growing trend to transfer responsibility for enterprise forms to Information Technology departments (who don't really understand forms), gradual downsizing and outsourcing of the Forms Management function within organizations, divergent, conflicting definitions of eForms, lack of understanding of forms and forms development requirements from many eForms vendors, and "scope creep" that afflicts many eForms development attempts in large organizations. Of course, there are other issues, but these seem to be among the primary causes of eForms program failures.

Many organizations have simply lost their expertise internally because they have either eliminated or downsized the function. In times of cost-cutting, forms departments make easy pickings to managements that are often removed from or unaware of the importance of professional forms management. Yet, every organization is dependent upon forms for virtually every activity and function. Forms provide structure to transactions, become the record of the transaction, and provide a legal record when transactions are challenged.

Badly designed forms cost an organization a lot of money. Various studies have been done over the years that clearly document this fact, yet badly designed forms still proliferate. We estimate (very conservatively) that, on average over the forms population of a company, badly designed forms cost \$1,000.00 per form in lost productivity due to errors, omissions, lost orders, poor customer service, and bad decisions resulting from bad data. It is not hard to prove this for any organization that doesn't have a professional forms management function that is properly staffed. When we do the math, a "typical" company with a population of 1,000 forms wastes more than \$1 million annually *due to poor forms design*.

As the number of pre-printed forms declines in an organization, even less emphasis is generally placed on forms management. Electronic forms are viewed as the responsibility of the Information Technology department and are typically viewed as front ends to databases. Forms are created using tools available, such as Microsoft Word, PageMaker, Lotus Notes, and Web development tools. They are frequently weak in function and utility. A study by Steve Weissman, of Kinetic Information (<http://www.kineticinfo.com/kimain.htm>) found that 65% of electronic forms in use were created using general-purpose software. Users wanted electronic forms but were unable to acquire forms design software so they used the tools that were available to them. As a result, electronic forms never evolved beyond the simple "fill and print" stage.

Where enterprise programs were developed, a common problem that prevented or limited their implementation was scope creep. Committees were formed to design the programs and every conceivable feature was added and specified. The projects became technology development projects and costs (and time to develop) made them prohibitive. As a result, nothing significant happened. Add to this the requirement that each user workstation needed client "filler" software, and growth of the industry slowed to a crawl. It simply costs too much and was too technically dependent.

We have had the opportunity to examine several forms programs and electronic forms opportunities over the years. It is quite clear to us that users want electronic forms that are readily available, easy to use, and cost-effective. One has only to visit State and Federal government websites and search for forms. An overwhelming percentage of available forms are simple "print-on-demand" and "fill and print" forms, with virtually no intelligence or online submission capability. That is the "state of the art" today. It doesn't have to be.

Let's look at the important elements and considerations for an effective and professional electronic forms strategy. We first start with definitions. There are no standard definitions that are generally accepted throughout the industry, so we will first define our terms and how we will use them in this paper.

## Definitions:

We define five “levels” of electronic forms. We further define the difference between “eForms,” “iForms,” and “pForms.”

### A. Levels of electronic forms

1. **Print-on-Demand (POD).** A form is designed electronically and made available to users. Users open the file and print the form for manual filling. Such forms eliminate the need for production, storage, pick ‘n’ pack, and delivery costs. They eliminate obsolescence costs, ensure the current edition is used, and eliminate or reduce bootleg forms. Cost savings are immediate and measurable.
2. **Fill-and-Print (F/P).** Simple mapping is applied to the form file that enables the user to fill it out online, then print and forward. Mapping is generally limited to tab order, some field restrictions, some masking, and special fields such as check boxes and simple drop-down selections. These forms provide the same advantages as POD, plus they are easier to fill out, more legible, and contain fewer errors.
3. **Intelligent Electronic Forms (IEF).** The next step is adding more intelligence to the form such as calculations, conditional fields, logic choices, logon access, hidden fields, and help messages. These forms are still self-contained, off-line and the data collected is not available to the enterprise.
4. **Enterprise-Enabled.** These forms are enterprise-enabled in that they employ email connections, database connections, secure access, intranet and/or Internet access, usage tracking, edition control, e-commerce connections, electronic signatures, and other enterprise features. They eliminate or reduce paper from the process, improve productivity, improve customer service, eliminate or reduce filing, and more.
5. **Forms-Based Applications.** IEFs and Enterprise-enabled forms are parts of a complete business application. These applications typically employ multiple forms and sub-forms in an integrated business solution. The forms software serves as a Rapid Development Application (RAD) environment, resulting in solutions that can be developed and deployed quickly. They require custom programming to build business rules and logic into the forms set. Full benefits are achieved, with no duplicate keystrokes, and the data are reused throughout the enterprise.

### B. Definitions

1. **eForms.** A form designed on a personal computer using software specifically intended for forms design; a form, the original artwork for which remains in electronic code format, rather than on artboards or negatives.
2. **iForms.** Electronic forms that run within a browser environment. These forms use HTML, JavaScript, and other technologies specific to browsers. They can be deployed on intranets or on the Internet.
3. **pForms.** An acronym for “paper forms”, it is generally used to describe form files that will be output to an imagesetter or plate setter for production on a forms press.
4. **vForms.** A computer-generated format, containing electronically displayed structural elements, static text, and graphics, which is processed on a personal computer or similar device using either specially prepared software or a web-enabled application. vForms exist as code until executed and displayed.

## Strategic Elements:

An enterprise-wide program generally includes plans for development, deployment, support, software standards, output strategy, management reporting, and cost-benefit requirements, including Return-On-Investment. These plans can vary widely between organizations and are dependent upon the organization’s existing infrastructure and standards already in place. Each element has specific considerations that should be decided and published as a part of the Enterprise eForms Guide. Looking at each element:

- A. **Development.** This addresses what department(s) and individuals are assigned forms development responsibilities. Frequently, anyone is authorized to create a basic design, describe the requirement and business process(es) supported, and submit the request to a professional designer. There is generally a process in place for creating new forms, revising existing forms, declaring forms inactive or obsolete, and

assigning control numbers such as form number, edition date, and retention requirements. Multiple versions of a form (State, Country, Language, etc.) may be required. Various approvals (legal, regulatory, marketing, etc.) may be required.

In addition, the Information technology infrastructure must be considered in areas such as database access, server scripts, networks, email compatibility, and more.

- B. **Deployment.** Strategies include email support, servers, forms portals, user access controls, user submission of filled forms, and security requirements (secure servers, e-commerce support, encryption, electronic signature support, and more).
- C. **Support.** This includes issues such as user training, help desk support, instruction manuals, user guides, and designer training.
- D. **Software standards.** These must be developed and supported. This generally becomes an important consideration for the design software and any user-required software (fillers, browser editions, and Acrobat® Reader edition, for example). Designer software developed to support the unique requirements for forms design is very important. We would never recommend using general-purpose software. Any business professional must be provided with the professional tools required by the profession. Forms designers need the same consideration.
- E. **Output strategy.** This defines how users will output the results of their fill sessions. One very early lesson learned is that users must be able to save and print their work or they simply will not use the electronic form. This is particularly true for the public doing business with your organization, but it also applies to employees. We generally think in terms of paper being unnecessary (after all, they are electronic forms), but this issue has stopped more than one electronic forms program. Users want what they want!
- F. **Management reporting.** Reports should include all statistics necessary to determine the extent of use, who uses the forms, development statistics, user requests for enhancements, and usage trends. Any program goals that are established should have concomitant metrics and reports established. One cannot manage what one does not measure!
- G. **Cost-benefit analysis.** This element is crucial to the long-term success of any program. For each form, the expected development and maintenance costs should be compared to the expected cost savings, including productivity improvements. A Return on Investment (ROI) should be calculated, including the expected payback period. If an acceptable ROI cannot be shown, the form should probably not be converted to electronic form.

There are several issues that arise in this important area. Many times, productivity benefits are not considered because they are “soft costs.” This is quite legitimate until it can be demonstrated how the benefits delivered can be converted to “hard dollars.” That function, however, is not within the control of the designer but is in the control of the form owner. We recommend using a method called Activity-Based Costing.

### **Workflow Reengineering:**

It really doesn't accomplish much simply to convert an existing form to IEF or Enterprise-enabled without first examining the workflow the form must support. Making a badly designed form electronic just means getting bad results quicker. Many organizations are guilty of limited work process studies because there is never enough time to do them. The result is inefficient processes. Ultimately, in an attempt to reduce costs, these organizations resort to mass layoffs. That quickly improves productivity (work divided by the number of workers), but such gains are usually very painful and short-term. Customer service can suffer, and the remaining employees are unhappy because work hasn't really been reduced - the survivors are working harder, not smarter.

The better way is to pay attention to work processes regularly. Study what really happens, challenge why processes are performed the way they are, and make them work more effectively. As business requirements change, modify the processes to accommodate the new requirements. Adjust resources to meet these changes quickly. Never let the organization become bloated by not paying attention to workflow. Then, layoffs can be averted or at least softened.

Forms lie at the heart of many business workflows. Understanding these processes is key to developing effective forms. Forms must fit the process, not the reverse. We recommend at least an annual review of each business process and the forms that support them. This staff function will generally pay for itself many times over.

### **Design:**

Effective forms design is a learned skill. Sure, anyone can add lines, rules, boxes, text, logos, and other objects to a screen (or piece of paper) and call it a form. But a form typically serves at least four major functions. Container design is only one of them. Care must be given to proper workflow, design analysis, needs of the data, the user interface, and other image requirements of a form.

In the early days of desktop publishing, untrained users acquired software, found that they had many fonts, colors, layouts, and other options available, and proceeded to create documents that closely resembled ransom notes. Just because these tools were available doesn't mean they knew how to use them. The same is true for forms today. Untrained designers create forms that result in the average ineffective use cost (\$1,000 per form, referred to above) that afflicts most organizations today.

Trained designers, analysts, and forms managers can generate substantial savings for most organizations. Equipping these professionals with proper design software is essential to achieving these savings.

### **Mapping/Programming:**

Mapping is the process of adding intelligence to the form by drawing fields and assigning properties to the fields that define how they will display, print, interact with databases, and perform other tasks. Most forms of software provide for field mapping. There is, however, a wide difference in the capabilities and degree of complexity supported.

Basically, forms software must be selected that provides the greatest range of mapping capability "out-of-the-box." Clearly, any functionality desired in the form that is not possible with point-and-click must be added with custom programming, which rapidly escalates the degree of complexity, time, and cost.

Field mapping needs to support a variety of capabilities, including naming conventions, masking, field types, required and hidden fields, conditional fields, and more. It is useful to develop a list of field attributes and compare various software tools on these attributes. Associates Group, Inc. has developed a spreadsheet of attributes for comparison purposes.

Another consideration is the programming language support provided within the selected software. Ideally, standard languages such as HTML, JavaScript, Visual Basic Script, Perl, and Java are supported and are extensible within the project. Avoid any proprietary languages or versions of languages.

### **Signatures:**

Signatures on electronic forms are still a very thorny area. Although the Electronic Signatures Act made electronically generated signatures "legal," there is a big difference between being legal and being accepted as evidence in court. Signatures must meet at least two tests:

#### **Is the signature the authentic signature of the person it purportedly represents?**

*and*

#### **Did the person signing intend to enter into the transaction represented?**

The document containing the signature must exactly represent the transaction as it occurred. This leads to the requirement for "non-repudiation." The information supporting the form must show and support a process that proves the transaction. This leads to a lot of workflow issues.

Generally speaking, forms development software must support electronic signatures in two ways, as a minimum. First, for internal transactions or transactions behind a user log in, the signature affixed must require the password to be entered again. Proper input of the password causes a signature to be affixed and the form's contents to "lock" so the signed form cannot be altered. Second, for the casual public (no login), the form must provide a link to third-party

signature technologies, such as VeriSign and Entrust. This enables secure signatures if the signer has a private key account. Of course, this is the problem. Most citizens do not have such signing capability. Consequently, we generally recommend that public documents requiring a signature be printed and signed manually and submitted separately to confirm data capture.

Electronic signatures can be accomplished with signature capture hardware that is integrated with the form. Again, very few citizens have this hardware, so it isn't generally practical for Internet-based transactions. It can be useful in retail situations where the signer is physically present.

Another alternative is to use a two-dimensional bar code that captures the variable data as it is input and prints a bar code on the form. The submitted form can be scanned and the variable data captured.

Enterprise strategies must identify the requirements for signature capture, both for internal forms and for external forms. This requirement will vary for each form based on the value of individual transactions and the potential for loss. Generally, signatures can be more casual for low-value transactions or where customer denial of the transaction is not likely. For larger value transactions such as insurance applications, or frequently contested transactions, such as beneficiary changes, the signatures must be more secure and non-repudiation measures are more important.

### **Routing:**

Many forms require routing to one or more persons after initial completion. Routing requirements are dependent upon the workflow and can vary significantly between forms and between organizations. The enterprise strategy should address the general requirements of the organization. Development software should be selected that most closely meets the enterprise requirement.

As a minimum, the form must be email capable. This means that submission of the form to a server generates an email choice for the user and launches the user's email, places a link to the filled form in the body of the email, and allows the user to address the email to multiple recipients. Recipients can open the form from the link.

The next level generates an automatic email to a pre-defined recipient (part of the form map). The recipient can be conditional or based on workflow rules associated with the form. Generally, such conditional routing requires the addition of a custom script. The forms software must support this addition of script and save it to the project.

Third-party workflow software is generally available to accommodate workflow mapping. Forms software that integrates with such software is usually preferred.

### **Security:**

Embedding security features into forms that are deployed to employees and to the general public will usually be a required feature for forms development software. These features should include encryption of data during transfer through the Internet, the ability to hide or show fields based on rules, controlled access to specific forms for specific users, field locking tools, password-protected fields, and version control features.

One of the common problems associated with such security is that the electronic form is generally held to a higher standard of security than the previous paper form. Ensuring that no unauthorized person can access a filled form is required, but when compared with the manual processing of paper forms, security should be designed to conform to the risks involved. It is generally more important to control payroll data than expense account data. The prime consideration should be a cost-benefit analysis.

Many security features are delivered and controlled by the server. Secure servers are required to process sensitive information such as credit card data, and security need not be duplicated at the form level. Databases that store form data should be encrypted and secure from non-authorized access. Generally, passwords should be obtained from the network logon and no separate logon file should be maintained. When regular dealings from the general public are required, a separate system for pre-registration of users, with appropriate logon and passwords, should be considered.

For each form or group of forms, appropriate levels of security should be defined, and software tools selected that support such requirements.

### **Database Connections:**

Generally, forms software must support connections to all Open Database Connectivity (ODBC) compliant databases. This would include most popular databases, such as Oracle, Sybase, DB2, SQL Server, MySQL, and Microsoft Access. Forms should be able to “read” (extract data for display) and “write” (place data into the database).

Database access is usually controlled tightly (for obvious data integrity reasons) by the organization and permissions must be assigned by the Database Administrator (DBA) before access is granted. Many times, direct access to a production database is not allowed, so data collection goes to an interim database for access by a custom application program.

An electronic forms strategy needs to consider how the organization will deal with the data collected by the form.

### **Storage/Retrieval:**

This is another potentially complex area of concern. Form data is generally stored separately from the form container. This requires that a specific association be created between the container and the data, so the transaction can be recalled and displayed appropriately. As the container is revised and new additions are created and deployed, the data display will change. Accordingly, all editions of a container must be maintained and accessible, with data mapped to the original container.

Forms software must provide this mapping automatically. It must associate the data and container edition.

As a matter of professional forms management, all editions of all containers are generally retained forever. As forms fall into deep archive status, the container may be removed from the active server, but it should always be available if needed.

One nice feature of forms development software is to provide an image of all completed forms to be created if required. This is generally a feature that is optional (selected by the designer) and, when activated, automatically writes an image of the completed form to a specified storage device. This image can be a flattened .pdf, a thumbnail, .jpeg, or .tiff.

Once submitted, a form must be able to be recalled by the originator, modified as necessary, and resubmitted. There are two basic options, determined by the designer: save as a new record or save over the previous record. Both options should be supported within the design software.

All electronic forms should support a print option. Users will want to print the filled form, print the container (unfilled form) and print only the data. If such print options are not supported, research shows many users simply will not use the form.

The same is true for saving the filled form. Users generally will not submit a form they cannot save. Additionally, many users prefer a draft mode save capability, where they can save partially filled forms until they are ready to submit. Forms development software should support such capabilities.

### **Other Considerations:**

Additional considerations for developing an enterprise eForms strategy include plans for workflow analysis, the need to support paper-based forms, electronic forms, and Internet-enabled forms from a single source file, how forms created outside the approved methods will be handled, interfaces to records management systems, search engines and search functionality, forms portals, and multi-network environments, managing obsolescence, user training requirements, help desk requirements, and IT Department support. Not one of these requirements is trivial.

### **Summary:**

An effective enterprise-wide electronic forms strategy is essential to the success of any program. Development of this strategy will involve consultation with all major departments to assess fully their requirements, needs, and preferences. It is important to remember that all forms need not be developed with all requirements intact. Do not over-engineer a form. Include only those features necessary to the workflow and avoid costly, over-technical solutions. Remember, paper is not the enemy. Paper is but another technology available to the professional designer. When electronic solutions become problematic, take the form to paper.

Most forms are already digital at some point in their lifecycle. The goal should be to keep them digital as long as possible. An effective strategy will enable an organization to do just that.