

Designing Fill-In Forms

Mayhew Chapter 5

Forms are Money

Suppose a system is being developed for 250 users, whose salary is about \$15/hr. Suppose there are 2-3 major fill-in screens which will be used 60 times a day, 230 days a year. Suppose some work could speed up processing by 3 seconds per form.

$$250 \text{ screens} \times 60 \text{ screens/day} \times 230 \text{ days/yr} \times 3 \text{ seconds/screen} \div 3600 \text{ seconds/hr} \times \$15/\text{hr} = \$43,125/\text{yr}$$

Organization and Layout

- Design and organize the form to support the task
- Organize groups of items related semantically by:
 - sequence of use,
 - frequency of use, and/or
 - relative importance.
- Keep the number of groups to a minimum, while limiting the size of groups to 12-14 chars wide and 6-7 lines high (within 5 degree visual angle).
- Use white space to:
 - create balance and symmetry,
 - lead the eye in the appropriate direction.
- Separate logical groups by spaces, lines, color, or other visual cues.
- Screens:
 - High-frequency users + slow system RT -- Minimize the number of screens.
 - Low-frequency users + fast system RT -- Maximize screen clarity.

- Keep related and interdependent items on the same screen.

Form Title -- (appears above URL in most browsers and is used by WWW search)		Background Color:
Q&D Software Development Order Desk		FFFBF0
Form Heading -- (appears at top of Web page in bold type)		Text Color:
Q&D Software Development Order Desk		000080
E-Mail responses to (will not appear on)	Alternate (for mailto forms only)	Background Graphic
dversch@q-d.com		
Text to appear in Submit button	Text to appear in Reset button	<input type="radio"/> Mailto
Send Order	Clear Form	<input checked="" type="radio"/> CGI
Scrolling Status Bar Message (max length = 200 characters)		
****WebMania 1.5b with Image Map Wizard is here!****		
<< Prev Tab		Next Tab >>

This image, from Webforms, simply hurts the eyes. Labels are not aligned to the fields they are associated with, causing the eyes to zig-zag around the screen as the user attempts to locate a field of interest. The choice of color to distinguish labels from editable fields further adds to the headache. Further, placing help information (will not appear on...WHAT?) in the labels just adds to the mess. Given that their [status bar](#) is too difficult to read, they probably decided that this was probably the next best place for it.

Caption and Field Design

- In western cultures:
 - for single fields, place caption to left.
 - for list fields, place caption above.
 - left justified above alpha lists.
 - right justified above numeric lists.
- Justify captions and fields according to user, task, and data type.
- Separate the (longest) caption (in a left-justified group) from its field by no more than 1-2 spaces (following the delimiter, for example, a colon). Separate one caption field group from another by three or more spaces horizontally, or by one or more lines vertically.
- Break up columnar fields or long columns of single field items into groups of 5 separated by a blank line.

- Provide distinctive field group and section headings in complex forms.
- Distinguish fields (i.e., contents) from captions (i.e., labels).
- Captions should be brief, familiar and descriptive.
- Indicate the number of character spaces available in a field.
- Indicate when fields are optional.

Input Formats

- Consider providing system completion of unambiguous partial input.
- Consider providing pop-up or pull-down menus for fill-in forms with many but well-defined entry options.
- Avoid complex rules for entering data in the various fields of a form.
- Provide meaningful (in field) groupings to break up long input formats (chunking).
- Provide defaults whenever possible. Allow simple (single key) acceptance of defaults.

Designing input data

- Make high-frequency inputs easy to express (e.g., y/n)
- Let the user specify the unit of measurement. Do not require transformations or calculations.
- Design meaningful input data whenever possible.
- Allow abbreviated input when it can be unambiguously interpreted (e.g., "y" for "yes")
- A system should be "case blind" when it really does not matter (e.g., "yes", "Yes" or "YES")
- Keep input fields short if possible.
- Do not combine letters and numbers in a single field.
- Avoid frequent shifts between upper- and lower-case characters.
- Avoid uncommon letter sequences.
- Do not require leading zeros.

Prompts and Instructions

- Provide prompts when use will be relatively infrequent, inputs must be formatted, and users are not working from a source document (i.e., help with syntax).
- Prompts should be brief and unambiguous.
- Place prompts to the right of fields, or in a MicroHelp line at the bottom of the screen/window.
- Provide instructions for navigation and completion on the screen or through on-line help.
- Place instructions in a consistent location across screens and make them visually distinctive.
- Use consistent terminology and consistent grammatical form and style in instructions.

PLEASE VERIFY THE INFORMATION BELOW AND MAKE ANY NECESSARY CHANGES

CC	1	Acct		-1	** PATIENT REGISTRATION **			Bal	
Remarks							Oth		
Pt Name							Guar		
DOB	S	SSN					SSN	DOB	S
Addr 1							Addr 1		
Addr 2							Addr 2		
Zip	City						Zip	City	
State	Home Ph()						State	Home Ph()	
Work Ph()	Ext	Misc					Work Ph()	Ext	
Ins	-					Ph ()	Note		
Rel	Insd					Emp	Name		
Pol ID	Group				Eff Dt				
Ins	-					Ph ()	Note		
Rel	Insd					Emp	Name		
Pol ID	Group				Eff Dt				
Rep	Date - Time	ID	Note	Text					
Chg Date		Chg Rep		Status		Def Stat		Accept?	

PATIENT SIGNATURE

HELP	CLEAR	SUBMIT
------	-------	--------

Navigation

- When a form is first entered, position the cursor in the most likely default position.
- Arrange field groups consistently with default cursor movement. Vertical groups are preferable to horizontal if cursor movement can be vertical.
- Allow forward and backward movement by field and within fields.
- Make protected areas on the screen completely inaccessible. Allow the cursor to rest only on user-editable areas.
- Do not use auto tab unless fields have fixed lengths and users are high frequency and experienced.

- Provide titles and page numbers or place markers on screens in a multiscreen form.
- Direct manipulation increases flexibility, speed, and ease of learning for navigation through fields.

Error Handling

- Allow character edits in fields.
- Place the cursor in the error field after error detection. Highlight the error field if possible.
- For independent fields, withhold error reporting until user request.
- Provide semantic and syntactic information in error msgs depending on user knowledge.

Tips for Data Entry Form Design

One of the most significant factors affecting data entry or forms processing cost is the design of the source document. Yet many organizations never consider the impact of the actual form design on data entry efficiency or accuracy until after they learn how much it will cost to process their documents. This is especially true when forms are designed by marketing or graphic design staff that place greater emphasis on aesthetics than on operational efficiency.

And not only do you have to consider the impact on keying or scanning, but what can really kill budgets is excessive time needed to prepare the documents for scanning or data entry caused by an unfortunate design flaw.

Some of the more significant factors are discussed below.

Constrained vs. Unconstrained Forms

Most people do not write very legibly, especially when they use cursive writing. Illegible hand writing and printing not only reduces accuracy; it also increases the time it takes for someone to key the data or verify keyed or recognized data.

On an unconstrained form there may be very little restrictions on how responses can be entered, which frequently promotes sloppy and illegible entries. The following are two examples of how fields might be set up on an unconstrained form:

Example 1:

Name _____
Address _____
Phone _____

Example 2:

Name:	
Address:	
Phone:	

If people are in a hurry when they fill out your forms (say while waiting in a retail store checkout lane), the above two formats will almost insure a high percent of sloppy and illegible entries.

The most effective way to promote legible hand print is to provide a lettering grid for each field into which a person would print one letter at a time into a single block. The following is an example:

First Name	J	O	H	N																
Last Name	S	M	I	T	H															
Address	1	2	3		M	A	I	N		S	T									
City		H	O	M	E	V	I	L	L	E										
State		P	A		Zip Code	1	2	3	4	5	-	1	2	1	2					
Phone		1	2	2	-	5	5	5	-	1	2	1	2							

Whether your data entry vendor is keying form paper, keying from scanned image, or using recognition technology, a constrained form will vastly improve the chances of data entry accuracy and efficiency.

Number Each Field or Question

This is not critical for simplistic forms involving only a handful of fields. However, it is absolutely critical when trying to enter the data from complex, multi-page questionnaires or surveys involving many different fields and many different types of fields.

For each new project requiring keying, a programming effort is undertaken to set up a “template” within our data entry system. The template must include a field for each field on a client’s form. When data entry operators are keying from scanned image (or keying from paper) the less they have to read, the faster and more accurate the data entry will be.

Imagine having to develop a data entry template for 27 page survey involving over 150 complex questions, each having several sub-parts – a total of over 600 potential values. If each question and sub-question is numbered, then all we have to put on the data entry screen are the question numbers, not the actual questions. Then the data entry operator merely has to read the question NUMBERS on the form (not the actual question) and key the answers into the corresponding

numbered field on the data entry screen. Without having the identifying numbers on the source document, data entry operators would have to read each question on the form and match it up with the corresponding field in the data entry program. In addition, the programming becomes much more difficult. We routinely decline such projects because of the high likelihood of error unless the forms are designed properly.

Use Check Boxes

Use check boxes (or radio buttons) to standardize answers and promote single-value entries. If your vendor employs recognition technology (OMR – Optical Mark Recognition) using check boxes can eliminate much of the manual data entry and promote accuracy. In this case, we recommend that you use a circle instead of a box, because people are more apt to fill in the circle (promoting recognition), but will usually put an “X” in the box (reducing recognition).

Give Each Checkbox a Numeric Value

If the forms will be keyed, we strongly recommend that each checkbox have a numeric value displayed next to it, representing the single digit the data entry operator will key if the box is checked. This not only cuts down on keystrokes, but also improves productivity and accuracy, because the data entry operator does not have to “read” and interpret the responses. The following is an example:

Highest Education Level:

- | | | |
|--------------------------|---------------|-----|
| <input type="checkbox"/> | High School? | [1] |
| <input type="checkbox"/> | Trade School? | [2] |

College? [3]

Although the above example is very simplistic, when we are dealing with complex forms involving hundreds of checkboxes, each having a different meaning, the task of “getting it right” can become daunting if the data entry operator has to match up the response on the form to the corresponding field on the data entry screen.

Avoid Stick-on Labels

Loyalty card programs and similar applications involve the use of a card number (or member number) which is assigned at the point of sale (or issue) and must appear on the enrollment form. To reduce printing costs, some organizations opt for having a sticker or label glued to the enrollment form. The unfortunate result of this decision is that it plays untold havoc on document handling, scanning, and data entry. For example:

Forms are difficult to handle, stack, store, and scan, due to the different thicknesses (i.e. thick where the sticker is, thin everywhere else – a stack of 100 forms measures 3 inches high on the end where the sticker is and 1 inch high on the other end)

- Forms stick together, and the bottom edges of some forms get stuck on the sticker of the next form – document prep time for scanning can easily double or triple.
- Stickers are applied upside down (images must be flipped if keyed, or numbers are not recognized if OCR is used)
- Stickers are applied in the wrong place (OCR does not find the number, or the data entry operator must look for it)
- Stickers fall off (customer not in database when card presented at store)
- Store personnel staple stickers to the form (staples must be removed)

The end result of using a sticker is that total cost is actually INCREASED, because the increase in document preparation, scanning, and data entry cost is far greater than the savings in printing costs.

As a result, we recommend that membership numbers be pre-printed on the enrollment forms. Although initial printing costs will be higher, this will result in a much more efficient and streamlined back end process, which will more than offset the higher printing costs.

Use Reasonable Field Sizes

The amount of space allocated to each field on your form needs to be reasonable. Too often, we see forms so small that the person filling in the information has inadequate space to print the information. A classic example is e-mail addresses. The average e-mail address is 20 characters long, yet you should allow room for at least 30-35 characters. The average name, including middle initial, is 15 characters, the average address is 16 characters, and the average city is 12. However, you need to allow much more room for these values. If you have a cramped form, the entries will be illegible, which will slow down processing and result in lower accuracy.

Use Registration Points on Documents to be Scanned

Most scanning software will use registration points on a document to align the image and make sure the appropriate fields are recognized, even if the document is fed into the scanner crooked. Registration points are typically simple geometric symbols, such as a dark square or triangle.

Use Check-Digits and Barcodes for Your ID Numbers

Whenever possible, use a check-digit as part of the ID number printed on your enrollment or registration form. A check-digit is a single numeric character whose value is calculated based on certain algorithm and used for the purpose of performing a mathematical check to ensure the accuracy of the data. A check-digit is usually the last character in the ID number. Whenever a check-digit is used in building your ID number scheme, we will incorporate the check-digit formula into our data entry system to validate the accuracy of the keyed or recognized number, thus promoting efficiency and higher accuracy.

Likewise, because we utilize barcode recognition in our scanning process, we are able to “read” the barcodes on your forms with a very high (near 100%) degree of accuracy. If the numeric values represented by the barcodes also include a check-digit, this will further enhance accuracy and efficiency in capturing the data.

Self-mailer vs. Envelope-stuffer

If your enrollment or customer response forms sent directly to Axion are mailed in an envelope, this drives up the cost of document preparation significantly. Our recommendation is to use self-mailers whenever possible.

Tri-fold, Bi-fold, or No-fold

No-fold is the best. When processing 100,000 documents for a client, having to unfold each one increases the cost of document preparation. We are more than happy to charge you for the extra work needed to prepare the documents for scanning; however, whenever possible, have the portion sent to Axion be a single, flat document.

Form Number (form ID)

When we are processing multiple forms for a client, each form must have its own “template” with data fields mapped to specific locations on each form. Using a pre-printed form ID number enables our scanning software to determine which form is being scanned. This makes the scanning process more efficient.

Keep It Short

People standing in check-out lines do not want to be slowed down by having to answer 100 questions – unless there is a direct benefit of doing so, such as a free subscription or entry into a contest. Too often we see a client’s double-sided, 8 ½ x 11 forms filled with optional “demographic” questions, yet many are returned blank except for the enrollment information at the top.

Effective use of forms on websites

Adam Baker's *theory* column for **March 1, 2001**

Summary: People don't like filling out forms in the real world, and especially not while using the web. Forms are complicated, distracting, and take control away from the user. That is, unless they're designed effectively.

Most people hate filling out tax forms, registration forms, ballots, and so on. In fact, all kinds of new technologies are being developed to speed up delivery of common information (like telephone numbers, addresses, names, birthdates, and social service numbers), and prevent excessive use of forms. Yet users are still forced to fill out forms, to accomplish all kinds of simple tasks, because forms are perceived as an effective means of gathering information. They aren't always effective for users, and certainly aren't always effective for the recipients of the information, either.

Some common user-perceived problems with forms

forms are long and complicated (they require a lot of cognitive activity, and interrupt [flow](#))

- forms frequently aren't clear, and discerning their precise purposes can be impossible
- forms usually force users to answer questions, even if the user perceives the questions to be irrelevant
- forms take control away from the user
- websites often yell at users if they don't fill out forms correctly

If users don't understand why they are forced to answer a bunch of questions, in a very particular way, to achieve an unclear purpose, they are going to have a negative experience. That might cause them to **simply leave the website**, or to miss out on the positive benefits that await them if they fill out the form correctly.

Some common owner-discovered problems with forms

- the information collected by the form has to be requested again by someone else within the company
- most of the information collected by the form is useless
- the form isn't helping the business achieve a set goal, or isn't applicable to any goals
- the important information users provide via the form is too thin (in other words, users aren't providing accurate or specific information)
- the form isn't being used as much as was anticipated
- few users are using the form for its intended purpose (if it has one)
- users aren't accessing the value-added portion of the website that using a form leads to

Forms are very mechanical by nature, but humans aren't. Because of interface constraints, it's much easier for most people to write a specific message than to deliver the same information by filling out a form (think of the repetitive mouse

movements required to select a birthdate from a complex popup menu widget). In many cases, if a user has reached a form, it's because they want something, or they want to communicate something to the people who run a website. They might have trouble trying to squeeze their thoughts into a narrowly-designed framework: a problem very similar to being forced to answer "yes" or "no" to a question when the answer should really be "maybe". If the message that the user needs to communicate is "maybe", then a human is much more able to process the information, and discover its meaning, than a computer is.

When are forms useful?

Obviously, forms are necessary for certain processes:

- collecting large quantities of statistical or otherwise quantifiable information
- surveys
- signup mechanisms
- login mechanisms
- search mechanisms

Feedback systems almost never require forms. Feedback forms are usually unnecessarily convoluted and frustrating. If a form is going to be used to direct information to a specific recipient or group, just indicate a specific email address to which the feedback should be sent. Give pointers with regard to what information users should provide, but don't force them to reveal information they are not comfortable with handing out to (potentially) complete strangers. Building trust and confidence is an essential part of positive user experience.

In circumstances where a form of some kind is necessary, it's best to keep it simple. In almost all cases, users want to be more effective, which includes getting things done hassle-free, and as quickly as possible. Therefore, **forms should have few controls, and shouldn't require excessive cognitive processing or repetitive physical movements** to complete.

Here are some common form design mistakes to avoid:

- long popup menus with lists of 150 nearly identical terms (such as lists of countries or professions; lists of two-letter abbreviations are exceptionally bad)
- asking for the same (or similar) information more than once

- asking for information that isn't relevant to the user's goals or task at hand (forget forms as a bonus market research tool)
- forcing users to fill out information (unless it's something absolutely necessary for maintaining user control like a user ID or password)
- forcing users to somehow communicate or build conversation with controls (no boolean buttons or complex search syntax forms)
- a form layout that puts might on will (borrowed from Alan Cooper: if most people just fill out one field and click "Submit:", put additional options after the submit button)

Tips for effective form use

I worked with a company that offered a very personal service to users. That meant that a registration system was required, and users would have to log in and log out. We ended up designing a service that users could browse before registration, but could only personalize after registration. The registration process was made as simple as possible: users were only required to enter a login and password to join. Additional information wasn't requested right away, but instead users were casually reminded to provide some more information, particularly their email address (for password retrieval). Otherwise, users only had to provide more personal information when it was necessary to help the user accomplish a given goal: for instance, users had to provide an address when they ordered a product for the first time.

The same service had a number of other simple forms for searching, logging in, and ordering products. In each case, we found out what was absolutely essential to the user, and how to most effectively design the form to keep users informed and in control.

Here are some guidelines that I kept in mind while working with the client to design the forms:

- use controls for the same purpose that they're used for in operating system interfaces (i.e. correct use of radio buttons and check boxes)
- collect information in bits and pieces as it's required from the user's point of view

determine the minimum information necessary for collection, so that the user remains in [flow](#) and doesn't become frustrated with the website

- only use a form if it has a very clear advantage for users over a less mechanical method, such as email
- use less restrictive controls (fewer radio buttons, but more text fields)
- make the form as "optional" as possible
- ensure that the user will never have to provide the same information twice
- test the form with the recipients of the information to make sure its contents are practical, and to eliminate anything that isn't practical

When considering use of a form, it's important to remember how much most people hate filling out complicated ballots, tax returns, registration forms, and surveys. Web forms are no more fun.