



INFORMATION  
ENGINEERING  
for the  
**21st**  
Century

BY JOHN BOWIE  
*Senior Member, Rocky Mountain Chapter*

**M**uch of the change we've seen in our profession over the past fifty years has been about doing the same things differently. First we delivered information to users in manuals, then in online help systems and, more recently, in embedded help. Despite these changes in information delivery technology, our roles remain fundamentally the same: We are, first and foremost, purveyors of technical information.

Information Engineering is about doing different things: about managing information rather than just delivering it, about designing information out of the user interface rather than building information into it, about requiring our products to learn more about our customers rather than requiring our customers to learn about our products. I hope you'll give it a try.

#### Fightin' Words

These words began an article I wrote for *Intercom* in May 1996:

Many, if not most, technical communication professionals waste their time providing services that no one wants or uses. Except in rare cases, they provide little or no value to the organizations that employ them, and fail in their efforts to communicate with the users of the products they support.

Back then, I challenged the STC membership to embrace Information Engineering as a new approach to technical communication, an approach focused less on rhetoric and information structure and more on managing and minimizing the information requirements foisted upon users by poor product designs.

Since that time, progress has been made. New advances in user interface design, such as right-click menus, tool tips, and embedded help, to name a few, have improved accessibility to product functionality and information. On the professional front, technical communicators continue to carve out new roles for themselves in sometimes resistant organizations.

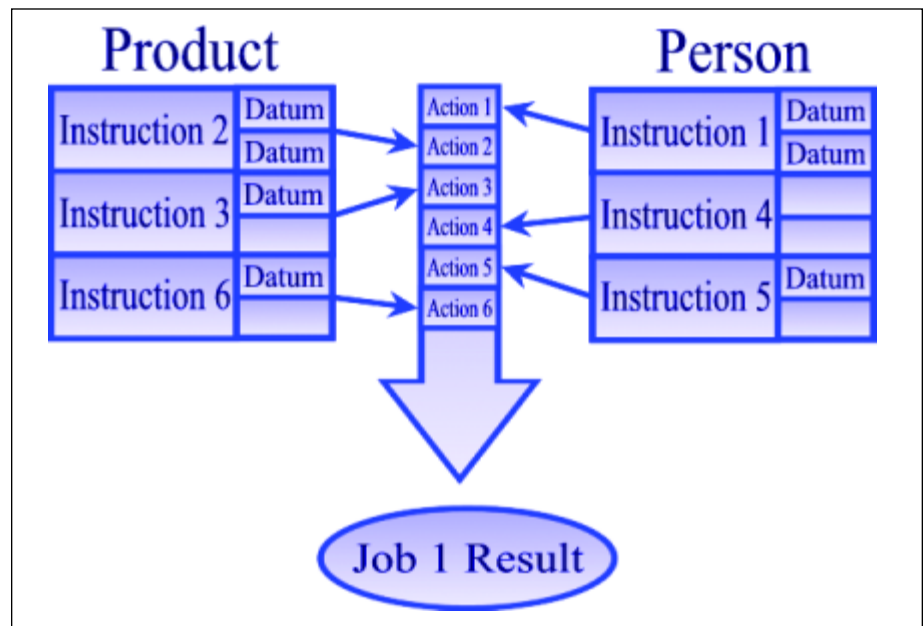
We've come a long way, but we have a long way to go. Manuals have all but disappeared from most consumer products. This would be a good thing if the *need* for manuals had disappeared also, but products are as complex and counterintuitive as ever. After more than a decade of development, it's time to admit that online help doesn't help. And Web-based information? Aside from its navigational nightmares, most Web-based support documents read like master's theses. The e-learning movement is gaining momentum, but what passes for e-learning today is often nothing more than page after

potential to improve the daily life of the average person, or more capacity to accelerate the development of the human race as a whole.

#### Root Problem

Every profession has a root problem—a fundamental need that makes the contributions of the profession valuable. Over the years, the root problem of technical communication has become buried under the billions of pages of information we've churned out. We've lost sight of our true mission; the means has become the ends.

Figure 1. Programming both products and people.



page of bulleted lists with a few pieces of stale clip art thrown in.

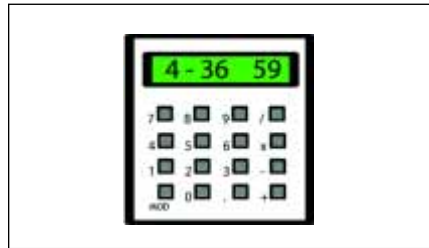
If you spend most of your time on any of these activities, you're not solving the problem that technical communication was intended to solve. You're wasting your time, your users' time, and your employer's time. Yes, I know you were *hired* to engage in these activities and spew forth these deliverables, but unless you return to your roots and rediscover the true nature of the problem you were hired to solve, you are doomed to spend your days shoveling schlock—and worse, contributing to the decline of world productivity.

Though I rail against technical communication as practiced today, I truly believe that no other profession has more

To rediscover technical communication's root problem we need look no further than our "users." Who among us has not been astonished at the lengths users will go to avoid reading our documentation? One of my colleagues convinced his company to spend hundreds of thousands of dollars to produce a beautiful multimedia training and information CD. He was frustrated that surveys and testing revealed that customers rarely bothered to use it. Therein lies the problem: Technical communicators want people to read their manuals, search their help systems, marvel at their multimedia, and take their training courses. They are focused on the *means*. Customers, on the other hand, will do almost anything to

BEFORE YOU  
CAN MAKE THE  
TRANSITION FROM  
TECHNICAL WRITER  
TO INFORMATION  
ENGINEER, YOU  
MUST LEARN TO  
VIEW THE WORLD  
DIFFERENTLY.

**Figure 2. User interface for a calculator watch.**



avoid reading, searching, and learning—they are focused on the *ends*.

These ends are called Job 1 Results. Job 1 Results are all the things your customer *wants* to achieve with your product. If your product is a portable CD player, the customer's Job 1 might be to listen to music while exercising. If your product is a cell phone, Job 1 might be to call business associates and clients while traveling.

Job 2 Results, on the other hand, are *requirements* imposed by the product that must be met before the Job 1 Result can be achieved. Job 2 Results are the means to the ends, and can be very undesirable. Tasks like installation, configuration, troubleshooting, searching, and reading are Job 2 requirements foisted upon users by complex product designs.

Job 2 requirements are the reason the technical communication profession exists. Our job has always been to bridge the knowledge gap between what people know and what they need to know to achieve the Job 1 Results. But the root problem of technical communication is not how to bridge the knowledge gap; the root problem is how to *close* it. Closing the knowledge gap is the primary focus of Information Engineering.

Now that we have rediscovered the root problem, we can evaluate our current solutions and discover new approaches that might serve us better.

### TC vs. IE

Information Engineering is based on the assertion that you can't always *write* yourself out of a problem you *designed* yourself into. No matter how elegant your prose, how well-structured your topic hierarchy, how engaging your learning modules, you cannot get around these universal truths:

- People don't want to read—they want to *do*.
- People don't want to search—they want to *find*.
- People don't want to learn—they want to *know*.

Yet look at how technical communicators spend their time: We know people want to *do*, but we try to solve the problem by writing. We know people want to *find*, but we approach the problem with search engines that yield dozens of hits per query. We know people want to *know*, but we approach the problem by teaching. Result: The deliverables of traditional technical communication are all but ignored by most users. A profession that produces deliverables that are ignored is obsolete. Yet the knowledge gap remains, and the root problem of technical communication is still unsolved. Fortunately for us, ours is the only profession with the focus and skills to do something about it.

**Table 1. The Information Inventory for the calculator watch.**

Instructions	Data
Press the MOD button three times	Location of MOD button Number of times to press (3 times)
Press an even-numbered button to select a field	Even-numbered buttons = select field Location of even-numbered buttons
Press an odd-numbered button to change the field	Odd-numbered buttons = change field Location of odd-numbered buttons
Press and release the MOD button	Location of MOD button

## Job 1 Results Systems

Before you can make the transition from technical writer to information engineer, you must learn to view the world differently. This change in mindset must occur before you can perform your job in a new way.

Technical communicators view products as fixed in design, while they view users as malleable receptacles for all the information they produce. Products do what products do, and user behavior must be reshaped to conform to the demands of the products. Users must learn to use the products; to do so, they must read the documentation, search online help systems for the answers to their questions, and commit time to various training and e-learning offerings.

Information engineers view users as fixed in design and attitude (or nearly so), while they view products as candidates for infinite design innovation. Users are what users are, and what they are is fed up with reading manuals, searching for information, and trudging through training programs.

Information engineers view the product and the user as components in a system for achieving Job 1 Results. Job 1 Results are achieved as the user and product components each perform actions; each action moves the system one step closer to the Job 1 Result. In order to act, each component must be programmed with instructions for performing its assigned actions. Product components are programmed with their instructions at the factory, but user com-

ponents often lack the instructions they need to perform their assigned actions. This is why technical communication exists: to program users with missing instructions, through the vehicles of documentation and training. Figure 1 (page 7) illustrates this concept.

The question before us is this: Are documentation and training effective solutions to the problem of programming users with missing instructions? If they aren't, the users cannot perform their assigned actions. If the user component fails, the Job 1 Result cannot be reached. And if the Job 1 Result cannot be reached, the entire system fails and the product is useless.

Herein is perhaps the primary difference between a technical writer and an information engineer: The technical writer accepts responsibility for writing user programs, but the information engineer accepts responsibility for the entire Job 1 Results System and works to ensure all components can function as intended.

Let's step through a technical communication assignment and see how we would handle it with the traditional documentation approach versus an Information Engineering approach.

### The Task: Setting the Time on a Calculator Watch

Let's say you're the technical communicator assigned to write the instructions for a calculator watch. The lead engineer for the watch hands you Figure 2 (page 8) and the following procedure for setting the time on this watch:

### HOWTO SETTHE TIME

1. Press the MOD button three times. The hours begin flashing, telling you that you can now set the time.
2. Press any even-numbered button to select which time field you want to set (date, hours, minutes, seconds). Then, press any odd-numbered button to change that field.
3. Press MOD again when you're done.

Now, if you are a traditional practitioner of technical communication, you might mutter something under your breath, but you'd then begin mulling how to translate this convoluted procedure into something more readable. You'd be focusing on the means of technical communication—writing instructions and creating illustrations—instead of the ends—managing the gap between what users know and what they need to know to achieve Job 1 Results.

### Step 1: Taking Inventory

Every software product has a set of minimum system requirements, such as processor speed, memory size, and operating system version, which must be met in order for the software to function. Most product development teams don't consider that every product imposes a set of *minimum user requirements* as well, which comprise all the information a user must know about the product in order to achieve Job 1 Results. Design complexity manifests itself as user information requirements. These requirements must be made explicit by taking an Information Inventory.

An Information Inventory captures *instructions* and *data* responsibilities that a product design assigns to the user. Instructions are the code the user executes to perform assigned actions; data are the bits of information that must be known before the instructions can be executed. Think of these as the software that must be installed in the user in order for the Job 1 Results System to function properly.

Table 1 (page 8) is the Information Inventory for the calculator watch.

The instructions are the lines of code the user must execute; the data are the parameters that allow each instruction to be executed. The next question to ask is this: How many people out there have

**Table 2. RAE analysis for the calculator watch.**

Instructions	Data	R	A	E
Press the MOD button three times	Location of MOD button	L	H	H
	Number of times to press	L	L	H
Press an even-numbered button to select a field	Even-numbered buttons = select field	L	L	H
	Location of even-numbered buttons	L	H	H
Press an odd-numbered button to change the field	Odd-numbered buttons = change field	L	L	H
	Location of odd-numbered buttons	L	H	H
Press and release the MOD button	Location of MOD button	L	H	H

this code preinstalled? In other words, how many users could set the time without referring to these instructions? The answer: Aside from employees at the watch company, very few.

The only salvation for this product now lies in the hands of the technical writer, who must find a way to program users with this missing information. A technical writer would dutifully write the instruction sheet and move on to the next project. An information engineer, however, would understand that most users will either lose the instruction sheet or reject these requirements outright and buy another watch.

The information engineer proceeds to the next step.

### Step 2: Rating Relevance, Accessibility, and Effectiveness

You would think that absurd designs would be recognized and rejected. The reality, however, is that they are very common. In most data-driven organizations, you must *prove* a design is faulty. You may be able to accomplish this with usability tests, but there is a much faster, less expensive alternative. You can perform a Relevance/Accessibility/Effectiveness (RAE) analysis on your Information Inventory.

For each instruction and its requisite data, ask the following questions:

- R** Is this information relevant to the Job 1 Result the user wants to achieve?
- A** Is this information readily accessible if the user needs to find it?
- E** Is this information presented in such a way that the user can easily understand it and use it to perform the assigned action?

For each question, rank the answer as either High (H) if the information has very high relevance, accessibility, or effec-

tiveness, or Low (L) if the information has very low relevance, accessibility, or effectiveness.

The scores for the calculator watch are shown in Table 2 (page 9).

The data requirements shown in bold italics are of particular concern, since each is both irrelevant *and* inaccessible. Why? Because memorizing the odd-even functionality has nothing to do with the Job 1 Result of telling the time. Furthermore, it's difficult to remember to press the MOD button three times in order to enter the time-setting mode. Users won't want to refer to the instructions for what should be a simple task.

When users are assigned irrelevant responsibilities, they get angry and frustrated. When users have to search for irrelevant information, they get even angrier. Irrelevant and inaccessible information requirements are likely to obstruct the user from setting the time, a critical Job 1 Results System failure. Job 1 Results System failures will cost the watch company money: in support, ill will, product returns, and low market share.

### Step 3: Redesigning the Job 1 Results System

Once you have discovered critical failures in the Job 1 Results System, you must redesign it to increase the likelihood of success. Here's how:

1. Eliminate or reassign irrelevant information requirements so the user doesn't have to deal with them. In the watch example, think of a design that would eliminate the requirement that users memorize the odd-even functions. For example, add a row of small buttons directly underneath the display, one each for changing the date, hours, and minutes.
2. If irrelevant information cannot be eliminated completely, then at least make certain the information is readily accessible. You could make the functionality of the buttons more obvious, eliminating the need for users to refer to the instruction sheet whenever they set the time.
3. Make sure that all remaining information is presented effectively. For example, you would not use words to

describe something that is best represented by a photo or drawing.

Your final design may not be perfect, but at least it will be improved. Every piece of information you can eliminate from the users' system requirements is a victory for the users and for you. Every piece of information that you can *deliver* to the users instead of making them *search* for it in an external source (manual, help system) improves the customer experience. And every piece of information that can be absorbed quickly and easily, without mental translation from one form to another, improves user communication.

You will thereby have evolved from a technical writer to an information engineer—a key contributor to your organization whose job is not simply to document complexity but to expose, analyze, and eliminate it.

### Restating the Challenge

In 1996, I closed my article with the following challenge:

Are you so entrenched in the manual business that you have forgotten your true mission? Are you passing up a chance to solve the core challenges of technical communication because you fear the solution might render your precious manuals—your business—obsolete?

Instead of worrying about finding new ways to entice people to read your manuals, why not invest your efforts in designing products that don't need manuals...or online help systems...or computer-based training...or telephone support? That's right—why not work yourself out of a job?

There's a better one waiting—if you have the courage to accept it.

The challenge is the same in the 21st century. If we live up to it, the years to come will be the best ever for our profession. **❶**

*John Bowie is the chief executive customer at Total Customer Experience Labs. He has spent the past twenty years encouraging technical communicators to take over the world. You can write to him at [john@icelabs.com](mailto:john@icelabs.com).*