



SOCIETY OF ACTUARIES

Article from:

CompAct

October 2008 – Issue No. 29

The End Users Justify the Means

Part 1: The General Public as End User

by *Mary Pat Campbell*

“Make something idiot-proof, and they will build a better idiot.”

I first realized I had an end user problem when I got a call from Denver, complaining that my annuity payment calculator spreadsheet wasn't working any more. As I listened, I thought: who is this person, and how did my spreadsheet end up in Denver?

Many times, spreadsheets we create for one purpose and one audience end up getting repurposed and re-audiened. In trying to make user-friendly spreadsheets, we can end up with very unfriendly users. In previous articles (To Err is Human; To Correct, Divine and Spreadsheet Check and Control: A Review) I covered the issues of preventing and correcting errors in your spreadsheets; here, I assume you have that part nailed, and are now readying to hand over your work of Excel art over to a bunch of people who will ruin it ... or at the very least, will put your spreadsheets through very hard use.

My antagonism with end users is a bit over-the-top, but often spreadsheet makers and end users seem to be in a war. The makers work very hard to get the moving bits working correctly, and the end users complain about the interfaces and that it doesn't deliver what they need. While it is very important to have the technical aspects of one's spreadsheet work, it is even more important that we take user needs into consideration.

Of course, not all users are created equal, and some people, such as those who have to maintain the spreadsheet (which could very well be us, the spreadsheet creators), don't seem like users at all. Users will have different needs depending on what they want out of

the spreadsheets, and they will have different capabilities. Here I start a series of articles on some of the major types of end users, and some general principles and specific techniques in dealing with them.

My intended schedule of articles:

1. The General Public
2. Numerate decision-makers
3. Auditors and testers
4. Maintainers

These articles are not intended to be exhaustive, and are based on my own direct experience as well as research into this area. At the end of this article are several references I've used, touching on different aspects of this subject, whether it's the visual design aspect or a systems perspective. I highly recommend other people write articles on this subject as well, or e-mail me at marypat.campbell@gmail.com with your ideas, so I can follow up on these articles.

First up: the toughest set of end users—the General Public a.k.a. the Great Unwashed Innumerate.

Granted, actuaries are not often called upon to write spreadsheets for use by the general public, though one may come close by being asked to make spreadsheet tools for the marketing or sales departments (OK, OK, many of them are numerate ... now.) This is not something to be proud about. How many times have you heard outsiders describe actuaries as subject matter experts, but horrible at non-technical communication? This is our own fault.

Actuaries are experts on many topics related to risk, and many people are poorly edu-

cated in this realm—we should be out there educating people on the various risks they face and how to manage or ameliorate them. Spreadsheets are but one tool we can use in this goal of communicating with and educating the public. As an example, I created a spreadsheet (which can be found on the SOA Web site at: <http://soa.org/research/pension/research-simple-life-calculator.aspx>. Editor's note: you must enable macros to use the calculator) to give people a better feel for longevity in retirement when most life expectancy numbers are quoted from birth (and you're only given expectancies, as opposed to percentiles as well). As of this writing, the first version of the spreadsheet is up, but I am assuming my second version is up by time of publication, which has some features described in the items below.

A. Is a spreadsheet appropriate?

Spreadsheets may be our general utility, but they're not always the best tool for the job. I like to think of Excel as a fancy calculator, allowing me to try different inputs to get a feel for a situation, or to let me enter information specific to my needs so as not to be distracted by irrelevant information.

If there is no interactivity, but a simple static display of information, then there's no need for a spreadsheet. This is especially true if you don't want anybody tampering with anything in the spreadsheet. Copy and paste into PowerPoint or Word if you just want to show a table of results, or a graph.

Tip: if you are copying a graph, make sure when you paste that you go into the *Edit > Paste Special* menu in Word or PowerPoint, and paste the graph as a picture as opposed

to an Excel object. I learned the hard way that when you blithely CTRL-C CTRL-V any graphic from Excel into Word, the entire spreadsheet is embedded within the Word document—for each graph you copy over. I crashed a few people's mailboxes by sending them my overloaded Word documents before I wised up.

If you really don't want people tampering with your information, it will be even better to get it into PDF format. Too often a graph in PowerPoint or Word can be moved to where its context is destroyed, and tables of numbers can be inadvertently—or purposefully—altered.

Another situation where a spreadsheet is not appropriate for a general audience is when you have sensitive or confidential information inside the spreadsheet. You may hide the sheets, lock the workbook, and password protect, but the security features of Microsoft Office are not totally secure. I have cracked many spreadsheet passwords with free add-ins you can find on the Internet. There is nothing secure about any Excel workbook. If you must use secret information to do your calculations, use another method, such as a Web interface to a secure server.

Tip: If you want to make a spreadsheet for general use where you strip out sensitive information from another spreadsheet, it's best to start out with an empty spreadsheet and copy over the parts you want rather than take the original spreadsheet and delete the sensitive parts. I'm probably being overly cautious here, but I can never be sure how much information Excel holds of previous versions of a spreadsheet; obviously it holds some information, or there would be no "Undo" functionality. In the life expectancy calculator, the

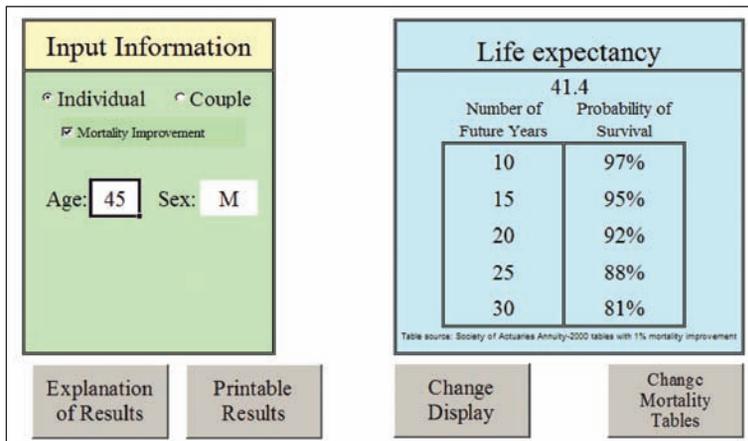
(continued on page 12)

original version was for use within TIAA-CREF and had some TIAA-CREF-specific mortality assumptions. In creating a version for the SOA, I started with an empty spreadsheet and copied over only a few of the mortality tables and nothing specific to TIAA-CREF.

B. Good interface design: think iPod

Apple has been the design leader of the 21st century, in my opinion (yes, I'm a Mac person, but I promise I'm not insufferable about it—much.). Where they have excelled is in clean, simplified interfaces, where a new user can pick up an object (such as an iPod), or open up an application (such as iPhoto), and get right to using it without having to muck about with user manuals ... and they're pretty, to boot!

When I started to design the interface for life expectancy calculator, I had Apple's design in mind. Here's a screen shot of the front page:



It's not the sleekest thing around, but it's very clear—large fonts for easy reading (except for the info in itty-bitty type, which is not going to be important to most general public users), clearly defined and separate input and output areas. There's no paper clip yabbering at you, asking what you'd like to do. There's a box for you to type in, buttons to push and an option to click on and off. Every button and option is clearly labeled with the least amount of

words. (I do think Input Information is clunky, though—I'd be happy to hear suggestions for my next version.)

Here are some general principles or issues I've considered in designing user interfaces for the general public:

i. Visual structure: Having well-defined areas on the page for various functions, whether input, output, explanatory or other. Also, always turn off the grid and row and column labels—only designers/maintainers will want to look at that.

ii. Uniformity of style: You don't want different elements clashing with each other. To go back to the Apple example, the iPod aesthetic is immediately recognizable, though it has expanded from the original model to include the Shuffle, the Nano, and the Touch—all with very different capabilities, but a commonality of design.

iii. Readability: Large fonts, dark text on light backgrounds, and space around different elements help improve readability of an interface. Edit wording down as much as possible—the fewer words used, the easier it is to read and the more likely it will be read. As Joel Spolsky of Joel on Software proclaims: "In fact, users don't read anything." A bit of hyperbole, but a good warning to prolix designers.

iv. Color: Excel has a limited default palette, which is why my spreadsheets tend to end up looking like an Easter window at Macy's. Color should be used with a specific purpose in mind. Here, I use color to set different regions apart in the spreadsheet. I used black text on a yellow background, as it provides one of the most eye-catching contrasts for reading (why do you think warning signs are yellow and black?), so that it would grab visual focus when this screen is first presented.

v. Font: Some fonts are easier to read than others. They all become less readable if you mix them up or lard on too many features, such as underlining, italics, boldface, all caps, colors other than black, and so forth. Generally, I use just one font—Times New Roman, a serifed font, at a sufficiently large size. Sometimes I use the sans serif Arial (the default for Excel most of the time) if I need to write something very small; serifed fonts do not work well at small size as the text-to-space ratio approaches illegibility.

C. Hide what isn't needed

I hide a lot of things in my spreadsheet. First, I turn off sheet tabs, the grid, and row and column headers. I hide input fields when you click on particular options and hide ridiculous results. For example, I do not show probabilities for survival beyond age 95, because the mortality tables beyond 95 are unreliable. But the end user doesn't need to know that. If you click between Individual and Couple, you will note that the age and sex input areas disappear and reappear. I do this by using VBA to set the background and font colors of the pertinent cells to the background color for the area.

If you want to see what lies behind the calculation, feel free to unlock the spreadsheet (I will talk about that below) and see what I have floating around. Calculating life expectancy is pretty simple, so there's not much to it. You will see some features I will talk about in later articles, which should be of use to auditors, testers and maintainers, but is of no interest to my target audience.

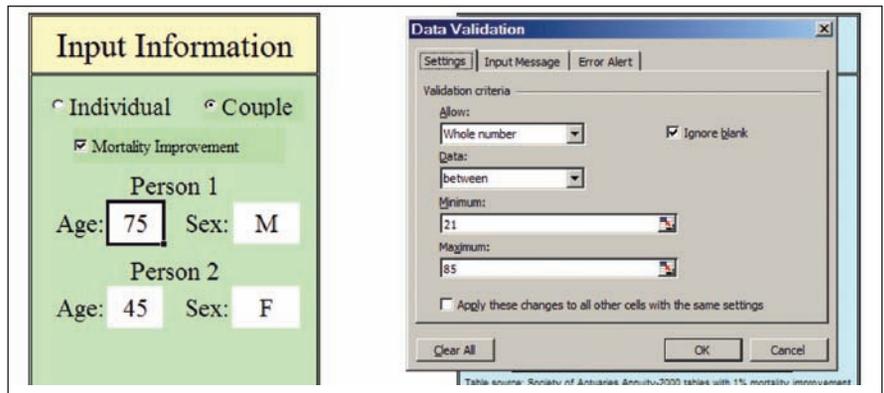
Hiding what the user doesn't need or want to see provides for a cleaner interface and reduces confusion. Also, in hiding the tabs and using buttons to force the user to navigate between two possible pages, I am controlling how the user navigates their way through the tool. I have done this many times for multi-

paged spreadsheets, to make sure that all input pages are gone through before a final calculation.

D. Constrain inputs and have sensible defaults

As a designer, I plan on whatever can be entered wrong, will be. You should have Data Validation on every input cell. Sure, the input prompt says Age, but just watch—someone will type in Forty-two in that field instead of 42.

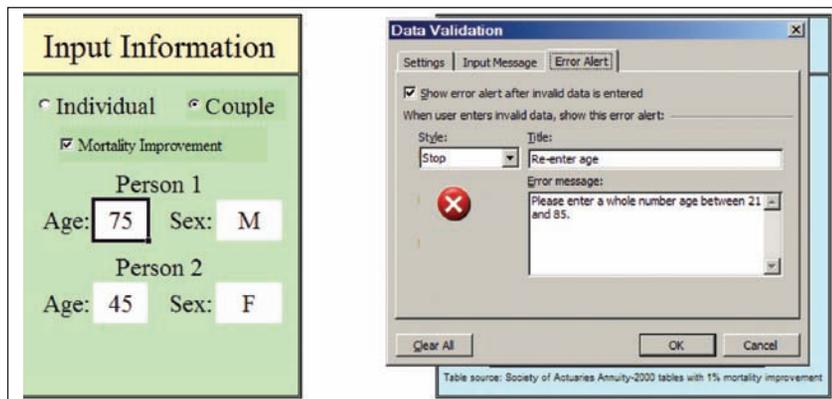
If you've never used Data Validation before, let me give you a quick walkthrough. I use two types of Data Validation in my spreadsheet. The first one relates to entering the age. To check this out, you need to unprotect the sheet (Go to Tools > Protection > Unprotect Sheet ... I don't use passwords). Highlight one of the age entry areas and pull down Data > Validation. You should see this:



I restrict entry of age to whole numbers between 21 and 85. There are some reasons for this—first, the point is to help people gauge retirement longevity or longevity from their current age. This is not intended for children; that explains the lower bound. The upper bound is due to the scarcity of data at high ages and lack of reliability in the extreme ages. Ideally, I expect the tool to be used for people age 45-70, but I expanded it to a more reasonable range.

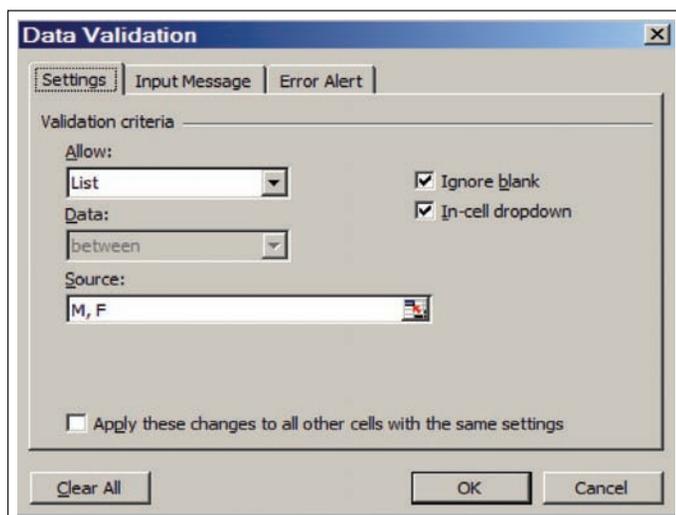
(continued on page 14)

When someone tries to enter something that does not fulfill the validation criteria, they get an error message. There is a generic one, but you should never allow that. You should always customize the error message to let the user know what you are looking for in that field:



Notice there is a third tab, which calls for an Input Message. I usually keep this empty, so as not to annoy knowledgeable users. I figure I should correct the users only if they have erred. You might think an Input Message would be appropriate when the input prompt does not supply enough information as to how you want the cell filled. I say that's a failure of design and the interface should be redesigned.

The other Data Validation I use is for sex:



The List criterion is a very useful option in Data Validation—if you have a small list of options to choose between. You can enter the list (as I did here), or refer to a range of cells (if you use cells from a different sheet, it must be a named range). The most useful feature of this option is the In-cell dropdown list, which automatically pops up when the cell is selected. The user will then automatically pick one of the items on the list. Note that I do not take this for granted—I still specify an error message, just in case the user is unusually creative.

While Data Validation is crucial to ensure that all inputs are such that calculation errors won't occur, another item to add is reasonable defaults to all input cells. I did not actually do this with my life expectancy calculator spreadsheet as the inputs are so simple. The reason to populate input cells with default values is to give the end user an idea of the range of values one is expecting to be entered. As well, many times people will not want to change defaults if they are not knowledgeable about the item entered (such as an expected interest rate, as an example).

E. Information at the point of need, only to the extent needed

Notice that I keep the explanation of what these numbers mean tucked away. I expect non-experts to need lengthy explanations and concrete examples to interpret the numerical results of my life expectancy calculator. However, I want to keep a clean interface and don't want to bother them with an explanation when they don't need it. There's a reason people got annoyed at Mr. Paperclip.

On the input page, I have a button simply labeled Explanation of Results. If you click that, an explanation of the numbers in terms of life expectancies and survival probabilities comes up—and it's directly related to what is currently in the input fields. Once one has

read the explanation once, that will probably be enough for many users, and they can go back to playing with the calculator without any distracting explanations.

Note, also, that I don't go deeply into the assumptions behind the mortality tables used (that comes later). When people are playing around with the simple interface, they just need to be able to interpret what the numbers mean. People generally don't like reading a bunch of text off a computer screen, so I try to keep the explanation as succinct as possible.

F. Printable results

The clean interface and simple explanations are all very good, but they don't print out

well. A computer screen is very different from a piece of paper. The empty space and large fonts that help with interactivity become low-density information when printed out. Also, the pop-up boxes with explanations do not appear in a print out.

It is a good idea to have a printable results page that has all inputs and outputs, as well as more detailed explanations than were on the input page. Sometimes your interface design will lend itself well to both computer screen and paper, but it's unlikely. The design concerns with a printable page are somewhat different from the computer interface—you will still want to have well-defined regions, carefully pick a font, and make sure the overall design is consistent. However, you will not

Simple Life Expectancy Calculator

Results for a couple:

Age of first person: 75 Sex of first person: Male
 Age of second person: 45 Sex of second person: Female

Mortality tables used: Society of Actuaries Annuity-2000
 Mortality improvement: 1% per year

Life expectancy: 45.2

| Survival probabilities for set number of years | |
|--|-------------------------|
| Number of future years | Probability of survival |
| 10 | 99% |
| 15 | 98% |
| 20 | 96% |
| 25 | 93% |
| 30 | 89% |

[Print This Page](#)

[Return to Calculator](#)

| Number of Future Years for set percentiles | |
|--|-------------------------|
| Number of future years | Probability of survival |
| 40 | 70% |
| 43 | 60% |
| 46 | 50% |
| 49 | 40% |

The calculations presented in this tool intended for educational purposes, as many people do not have a good feel for how long they may live in retirement. The calculation presented are based on public mortality tables which were developed with certain populations in mind, and reflect probabilities based on averages in large populations.

An individual's survival probabilities are dependent on factors such as family history, personal lifestyle (such as smoking or drinking), and existing disease (such as diabetes), and so on. One needs to take that into consideration when trying to consider how to use the numbers for their own retirement planning.

All probabilities and averages are computed taking into account your ages and sexes, using the Society of Actuaries Annuity-2000 mortality table, with mortality rates assumed to decrease by 1% per year.

The life expectancy is the average number of years until the last death in the couple. In this illustration, life expectancy is 45.2, which means on average you would expect the last person in the couple to die in 45.2 years.

(continued on page 16)

want to be using much color, as you can't count on your user having a color printer.

On a printable page, one can include far more results, and wordier explanations. The idea is that a person will print out the results and will be able to contemplate them at their leisure. Note still that I know the important part for most people will be the numerical results, so I used larger fonts for these items.

Users should be able to immediately focus on the meat of the page in looking at the printed product. People (who read English, at least) focus on the top left before anything else—it's a habit from decades of reading. So that's where I put the prime information.

G. Bulletproof your code

Simply put: the user should never see an error message, and you should never allow a user to do something that breaks the spreadsheet.

This is difficult to do, as you may think you have covered every possible screw-up, only to find that the user has found a new way to screw up. Hooray.

As noted above, one method of bulletproofing your spreadsheet is to add Data Validation to

input cells. Another thing to consider (which I did not do in my own spreadsheet) is to add catch-all error handling in your VBA (get to know On Error) and in Excel itself (get to know ISNA and ISERR).

A quick tip: make sure the spreadsheet recalculates after any change in inputs. My default for my Excel is manual calculation (as I'm often dealing with large spreadsheets or volatile functions), and your user may also

have such an option chosen when they open your spreadsheet. You can set the mode to automatic calculation upon the opening of the spreadsheet, but this could annoy the user (they may have good reason not to want automatic calculation), or the user may change it back to manual while the spreadsheet is open. You do not want the input and the results to be out of sync with each other.

H. Own your spreadsheet and protect the original

There should be contact information in the spreadsheet itself, and clear ownership of the spreadsheet. If you did your job well in following the rules above, you shouldn't have too many people contacting you about problems. Note in my life expectancy calculator, I have my name (people might have to know to look me up in the SOA directory, but if they search for my name online, they will have an easy time finding me). In addition, there's a link to the download page, which has the e-mail of an SOA staff member, as well as a link to download the current version of the tool.

The main reason to do this is to get feedback on your spreadsheet. There may be errors or missing features desired by users that you didn't catch, and allowing for users to contact you directly (or someone else who can contact you directly) will most efficiently get this information to you.

As well, you want to protect your original spreadsheet. In the method of dissemination shown here, the original is on a Web server, so the end user cannot tamper with it in any way. Sometimes, though, your spreadsheet will be sitting in a shared directory where anybody can use it. Some thoughts on handling this situation:

1. **BACKUP!** You should always have more than one copy of your spreadsheet lying around. I not only have my most impor-

I expect non-experts to need lengthy explanations and concrete examples to interpret the numerical results of my life expectancy calculator.

tant spreadsheets in a shared directory, but also in my hard drive and my personal network directory. If it's that crucial, I will even e-mail myself a copy, to have a copy sitting on the mail server.

2. Automatic copy: You can write some VBA code that will check the name/location of the file to see if it's the original. If it is the original, then it automatically saves the file under a different name. Here's some sample VBA code, which would be in the code under the VBA object ThisWorkbook:

```
Private Sub Workbook_Open()

If
(ThisWorkbook.FullName=OrigName) Then
ThisWorkbook.SaveAs Filename:=BaseName
& "TEMP"
End If
End Sub
```

OrigName and BaseName are defined as constants in a module. This is just an example. I think having the spreadsheet save a new copy every time it is opened is overkill.

As I said before, these ideas are nowhere near exhaustive. These issues come up in a variety of situations, so check out the references below as a place to start thinking about interface issues (generally, they are thinking

in terms of other contexts, but it still works for spreadsheets). ■

References:

Joel on Software—online series on user interface design for programmers. Starts with this post: <http://www.joelonsoftware.com/uibook/chapters/fog0000000057.html>

Wikipedia article: "Typography," subsection on Readability and Legibility: http://en.wikipedia.org/wiki/Typography#Readability_and_legibility

Color rules of thumb: <http://www.writedesigonline.com/resources/design/rules/color.html>

Jakob Nielsen—Ten Usability Heuristics: http://www.useit.com/papers/heuristic/heuristic_list.html (Jakob Nielsen is a usability guru. I highly recommend his publications: <http://www.useit.com/jakob/>)

Bruce Tognazzini—First Principles of Interaction Design: <http://www.asktog.com/basics/first-Principles.html>



Mary Pat Campbell, FSA, MAAA, is a vice president at The Infinite Actuary. She can be contacted at marypat.campbell@gmail.com.