From: www.cio.com

Cloud UI Design Mistakes to Avoid

– David Taber, CIO

December 15, 2011

I've written for years that it is impossible to make a product too easy to use. But the industry has proved me wrong, by making products that are so focused on easy that they encourage sloppy, unmaintainable system configurations. In the pursuit of something easy enough for mortals to use (and sales reps to demo), some cloud vendors are paving the way for a big mess a few months after deployment.

Focus on Good Looks!

Vendors are getting better and better at making UIs that are graphically pretty, with embedded animation and supposedly-helpful hints. But this often leads to "bright shiny objects" that obscure the things that the user is actually trying to achieve. Sure, the sizzle may sell...but it makes the product harder for the real pro to use. Beauty is only skin deep, even with UI skins.

Solution strategy: Push vendors for better navigation and "information architecture" in their products. Find out if the vendor has a usability group -- use that as one of your qualification criteria. Before you sign up, insist that you get a seat at future usability review or testing sessions.

Continuous Auto-Save!

Ever since Intuit used their continuous saving feature to clobber Microsoft Money, UI designers have been putting in continuous auto-save. It's nice, but with products that have some administrative complexity a simplistic auto-save is a guaranteed disaster. If the UI is going to be auto-saving complex rules, configurations, or parameters, it needs to also provide:

• The ability to backtrack (undo), preferably for all the changes made over several login sessions.

• Automatic logging of all changes in a setup audit trail, indicating who changed what when. Bonus points if the UI forces the person making the change to indicate why they're making it.

• The ability to snapshot a configuration and export it (preferably as an XML file).

• The ability to import those (XML) snapshot files, either for testing, disaster recovery or simple cloning.

Solution Strategy: Evaluate UIs for features along these lines, find out if there are third-party products (such as CVS) that could help, and pin the vendors down on their roadmaps before you sign.

It's so Easy!

Cloud vendors are innovating like crazy, and it is important that the features be usable. But over-focusing on the UI can mean that engineering has no time for programmatic access to features. And that means that a 'Great' product will be 'Not So Hot' when you try to put it to serious use:

• No way to add comments or rationales to a configuration? Do you really think you're going to remember the semantics of a pick-list or the meaning of that nested Boolean 3 months from now? As I wrote here, the first line of defense is self-documentation of each and every item as you configure it. But is that feature in your UI?

• No scripting? That means an awful lot of mouse-driving and serious difficulty in debugging.

• Lots of features in the UI that aren't available in the API? Tough to automate, tougher to integrate. Also means that you may not be able to retire that legacy application you were trying to move to the cloud for a lot longer than you thought.

• Audit trails, debugging messages, and error logs skimpy? Good luck trying to figure out which of your 347 rules is causing those extra leads to be created at 4 in the morning.

Solution Strategy: Think about what you need to achieve 12-18 months from now. How many people will be using the cloud system, how many things will be going on at once? Unless things are really uni-dimensional, you'll need to evaluate the cloud UIs for the issues above.

None of that Big System Complexity!

UIs that make it easy for the novice user often make it easy to generate complete chaos. Why? In the interests of user self-sufficiency, all the features are exposed at the lowest level (e.g., "all I need to do is send out this email..."). But there may not be a corresponding top-down view (e.g., "show me everywhere this variable is used to trigger emails"). Consequently, there may be dozens of rules or sequences that take almost the same action (or, are attempting in vain to take one action correctly), with no visibility or indication of overlap. This leads to conflicting scripts and a tough troubleshooting environment. Points off if the UI doesn't let you rename things once they are created, as this means you'll have to (1) come up with good naming conventions before you start doing anything practical, or (2) use the clone-and-destroy-the-original method to work around this problem.

Solution Strategy: Make sure you consider the needs of power users as well as novice ones, and

understand the real care-and-feeding requirements of that cloud UI over time. Don't be surprised to find out you'll need a half-time person (a trained one) to keep things running smoothly -- and budget accordingly!

David Taber is the author of the new Prentice Hall book, "Salesforce.com Secrets of Success" and is the CEO of SalesLogistix, a certified Salesforce.com consultancy focused on business process improvement through use of CRM systems. SalesLogistix clients are in North America, Europe, Israel, and India, and David has over 25 years experience in high tech, including 10 years at the VP level or above.

Follow everything from CIO.com on Twitter @CIOonline.