Raab associates, inc.

Building Usability into Your System Selection

David M. Raab
Raab Associates Inc.

A typical software selection project runs something like this: users define their needs; an analyst lists features related to those needs; an evaluation team researches features in the available products; and the product with the most features wins. It's all very organized and logical and moves steadily to a decision.

But this process ignores an elephant in the room. That elephant is usability.

It's not that users, analysts and evaluation teams are unaware of usability. In fact, usability usually ranks very high among their selection criteria. But usability is hard to measure, while features are easy to count. Small clear facts will beat big blurry realities every time.

Yet even a foggy reality can have painful consequences. Corporate data centers are filled with feature-packed software that goes unused because it's too hard to work with. The industry even has a term for it: "shelfware".

This doesn't have to happen. Usability can play a meaningful role in your purchase decision without greatly expanding the cost or time of the evaluation project. More than anything else, this requires a clear understanding of what usability really is.

The critical insight is this: *usability is always relative*. That is, usability can only be measured in the context of specific users, tasks and circumstances. A system that is easy for a skilled technician may be impenetrable to a beginner. A system can be good at one task and poor at another. A system that works well after elaborate set-up may be a disaster when quick changes are needed. This is why published reviews can't include a usability score as easily as a feature checklist. Features are the same for everyone; usability depends on the user.

Once you recognize that usability is relative, the process for measuring it loses its mystery. First define your particular situation: Who are my users? What tasks will they perform? Under what circumstances? Then, rate how well the software will work under those conditions.

The starting point is tasks. These are derived from your needs analysis, which first identifies the system outputs (marketing campaigns, closed sales, handled support requests, etc.) and then

describes the processes that create those outputs. The steps within the processes are the tasks that users perform.

Most systems will have several types of users. Technicians set things up; administrators manage them on a day-to-day basis; end-users perform a limited range of repetitive operations. You need to identify the relevant groups of users in your particular organization and then determine which group will perform each task.

The next step is to assess how easily each group will perform its tasks. Ideally, the group members will look at the potential systems and provide their own answers. This can be done by watching a demonstration or by working hands-on with a test system. Sometimes, someone else can assess usability on the group's behalf. A call center manager might judge how easily her agents will work with the system.

Remember that the same task may be performed in different systems by different users. For example, some products allow end-users to add a custom field, while other systems require a system administrator or database technician. Also distinguish the effort to create something the first time from the effort to reuse it. For example, a campaign template may be hard to set up but easy to modify. Building a detailed understanding who does what isn't easy, but it gives very important insights into how independently end-users can operate and what demands the system will place on each group of users. (Note: a spreadsheet to help capture usability assessments is available in the Resources section of www.raabguide.com.)

Once you're determined the users and effort associated with each task, you have the data needed to compare usability across systems. If the winner isn't obvious, have the project team build a comparison matrix with ratings for each task for each system. You can weight the scores and calculate a summary rating for each product, but the actual scores are less important than the discussions that go into creating them. These discussions are how the team builds a shared understanding of the real differences between the products.

This table highlights differences between feature- and task-oriented selection processes:

Feature-Oriented Process	Task-Oriented Process
1. define business needs	1. define business needs
2. define processes to meet needs	2. define processes to meet needs
3. define required features	**3. define tasks within processes**
	4. identify users for each task
4. identify candidate systems	5. identify candidate systems
5. assess features for each system	**6. assess user effort for each task, for each system**
6. compare, rank and select systems	7. compare, rank and selection systems

As the table shows, the processes actually share most of the same steps. Where they do differ is whether the evaluation team will spend its time looking at features or analyzing business processes. You'll need to understand those processes eventually in order to implement the system, whereas research into features will never be looked at again. So the task-oriented approach will save time in the long run by giving you a head start on the implementation process.

Let's step back a moment. What really matters about a system is not its features, but how well it meets your business needs. By focusing on tasks and users, the task-oriented approach lets you measure directly how each system will fit into your business. It avoids both the classic error of ignoring usability altogether, and the more subtle mistake of treating usability as a dimension of its own. Usability is part of the complete picture of a software product and must be evaluated in the context of the whole.

* * *

 www.raabassociatesinc.com David M. Raab is a Principal at Raab Associates Inc. and author of the *Raab Guide to Demand Generation Systems* www.raabguide.com. He can be reached at draab@raabassociates.com.