Help Users Love Your Apps: Create Custom Dashboards

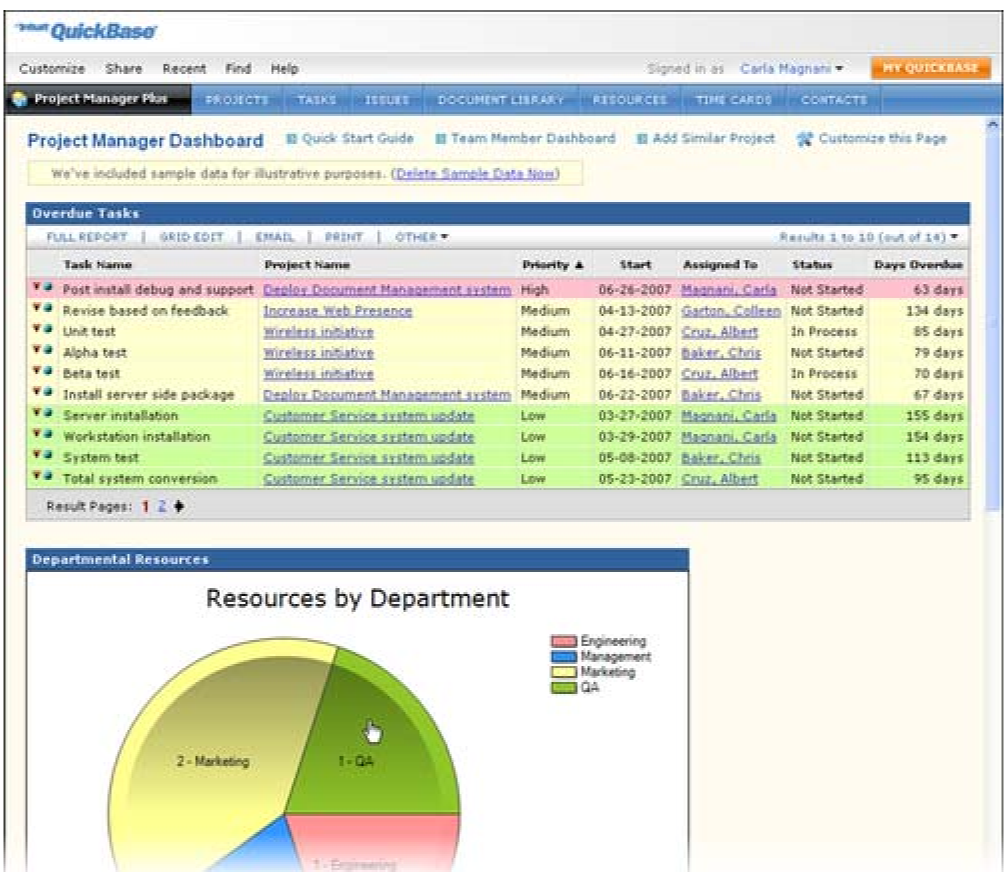
Would you like to design applications that your colleagues will beg you to use? Not long ago, Peter Fearey wrote an excellent post entitled Helping Users Love your Apps. Today's post is the first in a series of follow-ups that will expand on his thoughts. If you implement all these tips, working in your application will be a pleasure and not a chore.

What's the first page your users see? It's the Application Dashboard, which is essentially the "home page" of your application. By creating a smart and informative dashboard (or many) you can save your colleagues loads of time. In fact, I've seen applications where users can tackle all their jobs from this single screen.

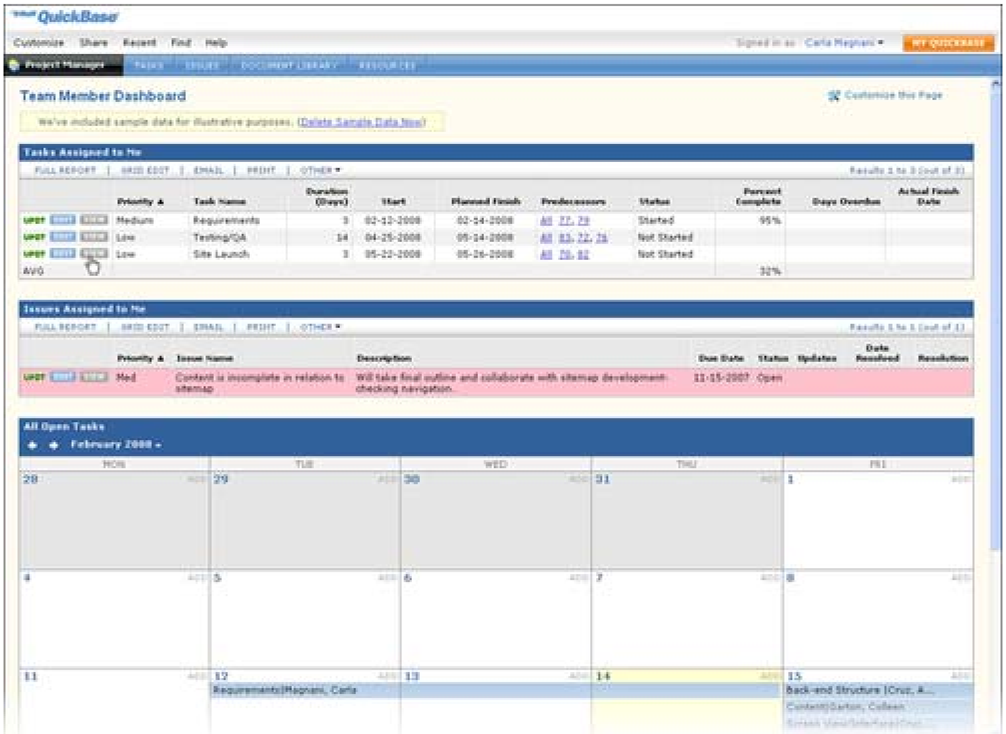
Customize by Role

Start by asking yourself what each user needs to see. Usually, this need breaks down by role. For example, executives may want to see overall status on many projects, but an individual staff member is only interested in what tasks he must perform today. In this situation, create a different dashboard for each type of user. QuickBase lets you designate one of these dashboards as the "home page" for each role. (You can even create multiple dashboards for each role, which you'll read about in a minute.)

For example, show this dashboard to executives:



and show this dashboard to team members:

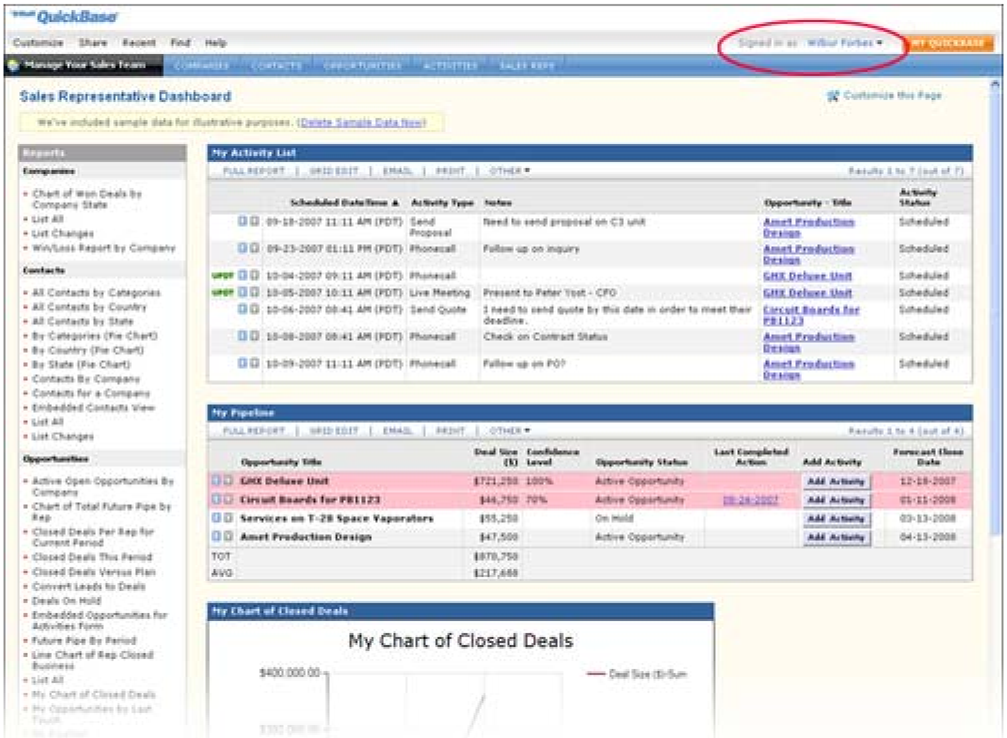


Customize by User

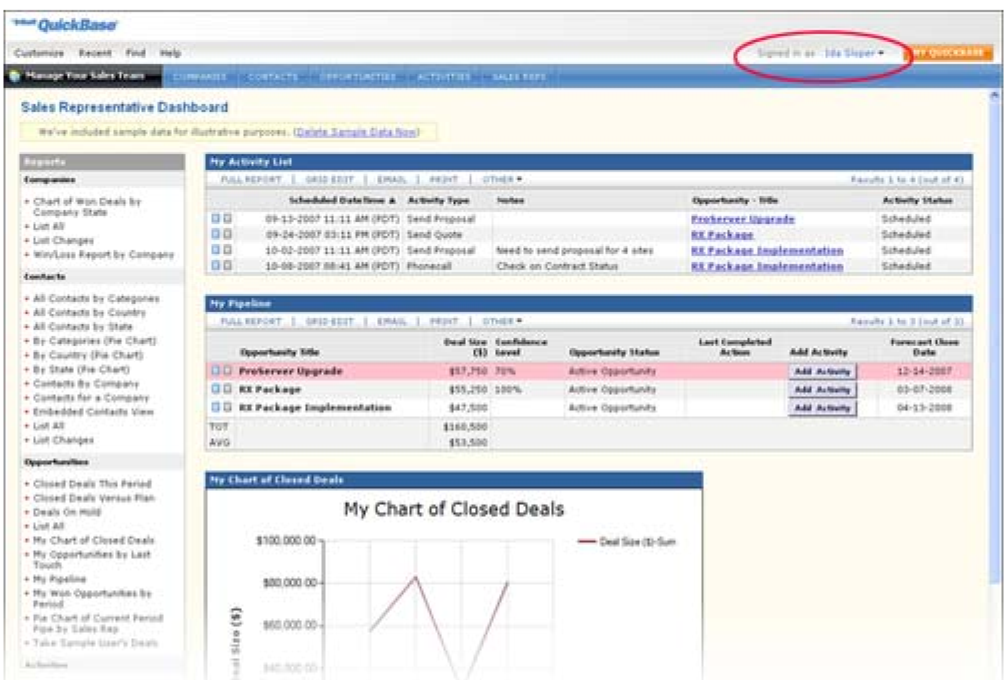
Customization doesn't stop at roles. Someone recently wrote in and said "I see how to customize by role, but how do I create custom dashboards for each user? I don't want to see my colleague's to do list, only my own."

To accomplish this, just create a report that automatically shows the current user only those items that pertain to her and embed that report on a common dashboard. The result? - A dashboard that's custom fit for each individual user.

For example, Sales Rep Wilbur Forbes would see this:



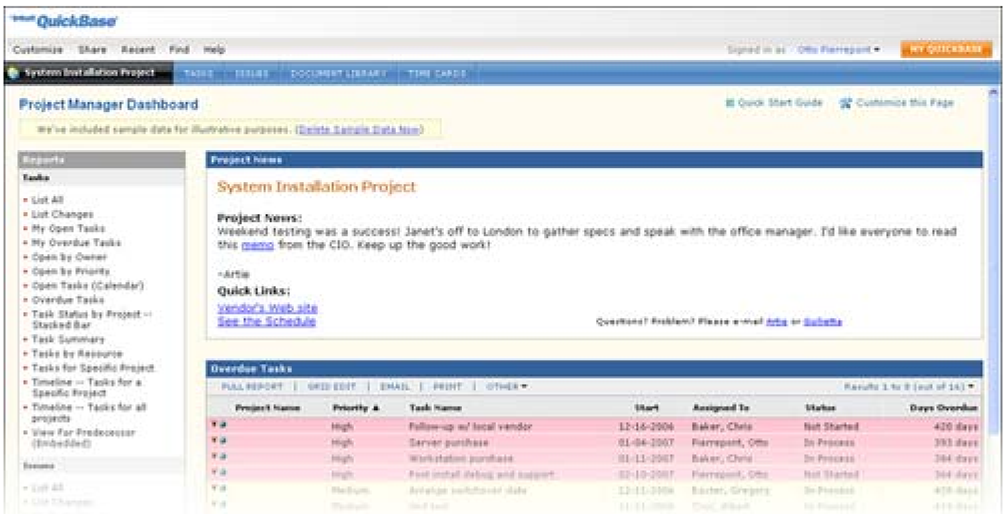
and Sales Rep Ida Sloper sees this:



Although they are both looking at the "same dashboard," each one sees only his or her own activities and opportunities. Embedding these user-specific reports is a great way to cut through all the data and show your users only those items that interest them.

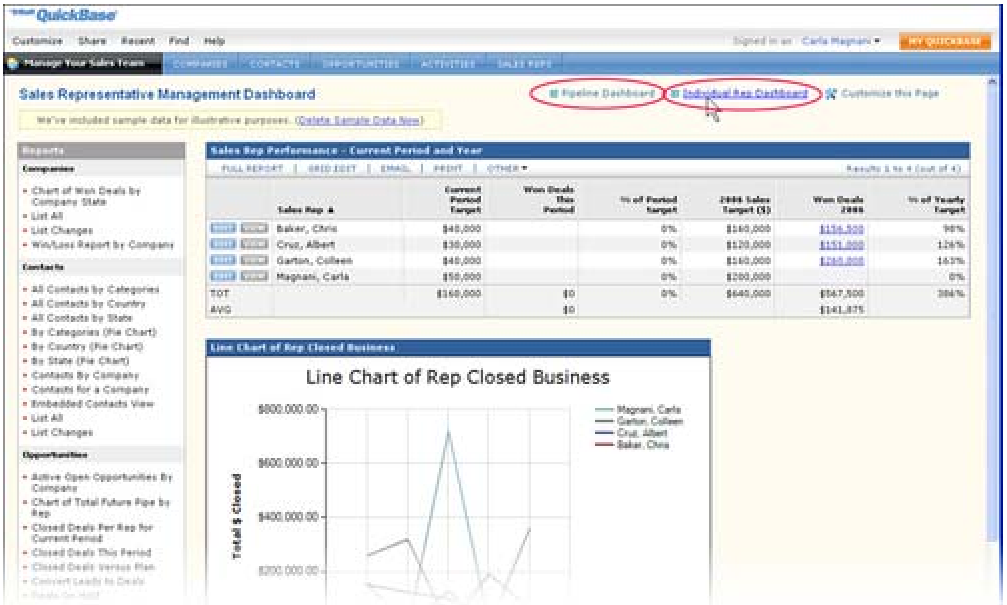
Display important messages and guide users

Does your staff have trouble communicating? Help everyone out by telling them stuff they need to know. Do so directly on the dashboard. [Text sections] let you post messages and/or instructions. You can format these missives any way you wish--in huge red font or discreet tiny gray font--whatever's appropriate. You can even include hyperlinks to pages in your application or out on the Web.



When one dashboard's not enough

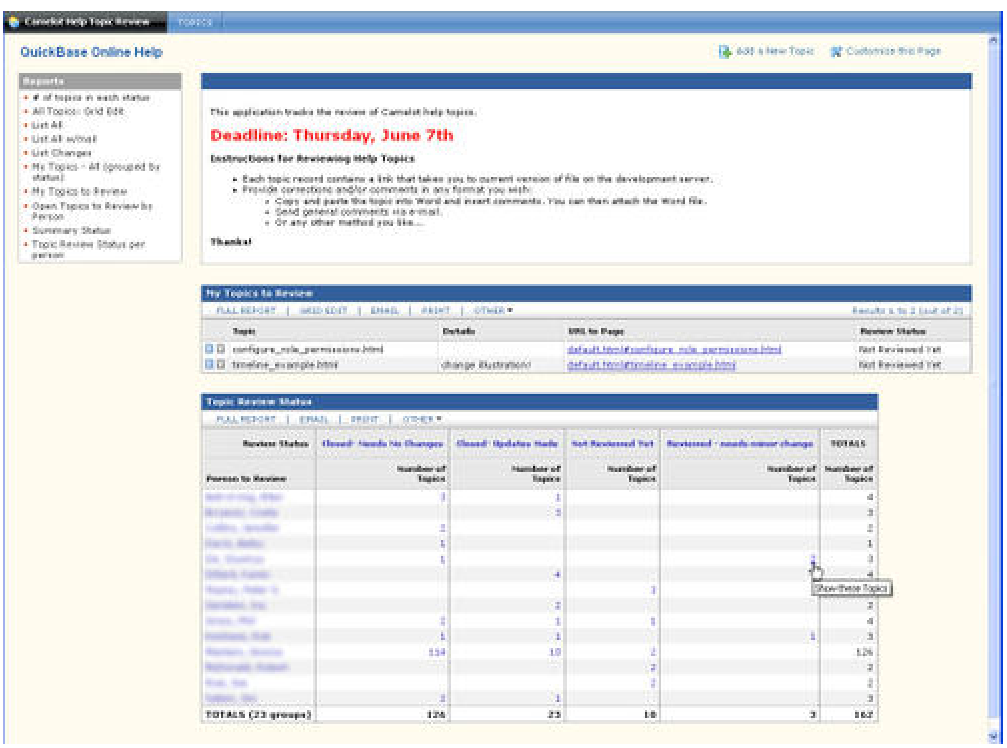
If necessary, you can create multiple dashboards for a single role. For example a sales manager may want one dashboard for pipeline management and another for sales rep management. Once you create each dashboard, [add custom buttons to the top of the dashboard page] and users can quickly switch between dashboards.



This dashboard shows the manager how sales reps are performing. Using the circled links, the manager can quickly switch to other dashboards to monitor the sales pipeline or see detailed reports on individual reps.

Show users how their progress compares with others

Finally, I'll show you a dashboard that I designed. This one tracks internal review of help topics. It contains some of the elements you've just read about, like a text section with instructions and a user-customized report that shows the current viewer only those topics he or she must review.



One new wrinkle is the table on the bottom. It shows overall progress--namely how many topics each person has reviewed or not reviewed. (Names are intentionally blurred to protect the innocent.) Not only can my colleagues see if they've fallen behind others, I can see what topics require edits. A single click on a number in the *Reviewed - needs change* column opens a report with all the feedback someone's got for me.

The goal of this dashboard was to guide users, spur action, and speed tasks. I used custom reports and useful shortcuts to make reviewing topics and giving feedback as easy as possible for folks who were, in fact, doing me a favor. Do the same for users of your applications. The time you save may be your own.

Understand what users need

Finally, as Peter emphasized, listen, learn and evolve. Ask you users how they like your application. Stand behind them as they try to use it to do their work. Last week, David Pogue stressed the importance of Designing What's Right for Consumers. Your users are your consumers. Craft your application around them. One of the greatest things about QuickBase is that you can keep improving your application and its dashboards, even after everyone's been using it for a long time.