SLS Usability Test Results

Chris Jackson
August 19, 2002

Subject: Software Licensing System (SLS) Paper Prototypes
Test Participants: Debbie Gordon, Tim Morrison, and Willard Stinson
Test Dates: August 12, 13, and 16
Design Team: Tina Jenkins, Joyce Metz, Chris Alexander, Chris Jackson

Summary

Using the SLS use-case model, the design team performed task analysis, constructed abstract models, drafted paper prototypes, and performed usability testing for several generations of the paper prototype with three participants: one from Sales, one from Professional Services, and one from QC. We could perform two more tests (an industry guideline suggests that testing an interface with five participants finds 80% of an interface's problems), but recognize the time constraints of the project.

This version of the prototype organizes the main parts of the software into three modes: Search, Edit, and Create.

- Search mode is what is displayed when the user logs on to SLS. Search mode displays simple search fields and a link to advanced search near the top of the window. The rest of the window is a table display of search results; this table defaults to displaying all license agreements to which the user has access.
- Create mode is a five-step wizard to walk users through entering the information necessary to create a license agreement. The wizard opens in a new browser window, so users can refer back to existing agreements while creating a new agreement.
- Edit mode is a tabbed interface within the main window; most tabs are similar to the wizard steps in Create mode. The Licenses tab, however, is available in Edit mode but not in Create mode, because it is a view-only display for existing contracts.

This version of the prototype addresses the usability issues listed below.

Terminology

License Agreement: There was some confusion between "license," "license agreement," and "contract." In SLS, a *license agreement* is the composite database record for all of the information associated with a single sales contract or demo version. A *license* is an activation of the terms of a sales contract for a single end user or machine installation. A *contract* is the sales agreement document between ENA and a customer.

Admin: In SLS, an admin customer contact has access to the customer view-only version of SLS. In most cases, an admin contact will be responsible for distributing licensing information to end users.

Interface Issues

Usability testing found the following issues with the SLS interface:

Module	Issue	Action
All, Required Fields	The design team had a question about the obviousness of “required field” notification.	Move asterisks for label text to the left side of the labels, so that (for most labels) the asterisks sit aligned in the left margin (making it easier for users to scan the page and see what is required).
All, Document Linking	The participant brought up the idea of linking from SLS to the actual contract document, or at least using the same identification number for the license agreement and the signed contract document for legal reasons.	There may be a future enhancement for linking to the contract document. In the meantime, the design team will recommend to the persons in charge of sales contract workflow append the Contract ID value generated by the SLS to the contract document.
All, Terminology	The participant said that he was somewhat confused by the different terminology of “license” and “contract.” He wasn’t sure if the software was used to create actual contract documents.	While the terminology could be explained by a Help file or by a Training video (accessed from the Help menu), the design team recommends that SLS use “license agreement” exclusively, except for the field label “Contract ID,” which may be used to refer to the contract document.
Change Password	Participant suggested changing “New Password (again)” to “Confirm New Password” and changing “Save” to “Save Password”.	Make these changes.
Change Password	Participant asked whether or not he would be told the field restrictions for passwords.	Add onscreen help text stating restrictions.
Communities Tab/Step	When adding an existing community to Selected Communities, the participant wasn’t sure if she should click the Add Community button.	Change button label from “Add Community” to “Add New Community” or “Create New Community.”
Community Dialog, Add New	The design team observed that there needs to be a way to select the parent item when adding a community.	Add a required Parent text field with a Select button that opens a pop-up containing a tree list of communities.
Contact Dialog, Add New	Participant suggested adding “Admin” to the Type dropdown list.	While it would make the interface a little cleaner, we can’t do that because most admin users will also

Module	Issue	Action
		have a type, such as Technology Coordinator, associated with them.
Contacts Tab/Step	The participant pointed out that there could be two contacts with the same name in the Available Contacts list. He wasn't sure how he would tell who was whom.	Since the Contacts data is filtered by the Customer automatically (new contacts are associated with customers behind the scenes), chances of multiple contacts with the same names are slimmer. Also, the user can select the contact and click Edit Contact to see who is whom. The design team determines this to be a non-issue.
Create New LA Wizard	The participant wondered how she would look up and copy information from an existing license agreement to a new license agreement she was creating.	The New License Agreement wizard should open in a separate modal browser window. The new browser window will not have the main navigation links (to avoid windows on top of windows on top of windows).
Edit Mode	The participant wasn't sure if he had to click Save before moving from one tab to another.	Provide Warning pop-up when a user tries to leave a tab without saving modified data.
Edit Mode	The participant wasn't sure how to save data changes.	Move Save button to lower-right corner of Edit screens (similar to where Next button is in Create mode).
Email Tab/Step	The design team realized that there is no way to notify admin contacts of license agreement information.	Add an Email tab/step that allows the user to create a boilerplate email message in his or her email client that contains the required information for the license agreement (Contract ID, Download URL, and so on).
Expiration Tab/Step	The participant was not entirely clear on the difference between license expiration and contract expiration in the Expiration step of creating a new contract. We discussed possibly separating the demo-purposed parts and contract-purposed parts into separate halves of the screen. A step further would be to create different wizards for demo licenses versus contract licenses. The different wizards	After discussion, the design team decided to split the Expiration step/tab into two sections controlled by a "Demo License or License Agreement?" radio group.

Module	Issue	Action
	wouldn't have to look that different, just collect the information appropriate for each purpose.	
Expiration Tab/Step	The participant suggested having an editable dropdown for expiration schedules, with default schedules to choose from. He also suggested measuring the expiration schedule by percentages rather than days; for example, expiration notification could be occur at halfway through the contract period, three-quarters of the way through the contract period, and so on.	The design team decided that this may be a future enhancement.
Expiration Tab/Step	The numbers in the Notification Schedule field are not defined.	Add "days prior to expiration" after the Notification Schedule field.
General Step	The participant suggested that the Effective Date could have a Date Picker button.	Add a Date Picker button. Also, we talked about having the field default to the current date for ease of use, but we decided that this "feature" could breed bad data (by someone forgetting to change the preset data). Better to leave the default txt as "mm/dd/yyyy".
General Tab/Step	Participant suggested making the Product field a dropdown.	Make the Product field an editable dropdown.
General Tab/Step	The participant asked how he would create a new customer.	Add a Create New Customer option to the top of the dropdown list. This brings up the Create New Community dialog, labeled "New Customer" and without the Type dropdown.
General Tab/Step	In the General step of creating a new contract, the participant wasn't sure what to do to create a contract with an unlimited number of installations.	Add Help text to the screen, something like, "leave blank for unlimited installs," placed to the right of the field. Ditto for unlimited user licenses.
General Tab/Step	Participant wasn't sure what the difference was between "Customer Notes" and "Internal Notes." He was not aware of the limited, view-only version of SLS for customers.	Change the field label to "Customer Viewable Notes."
General Tab/Step	The participant suggested making the Max Installs and Max Users fields editable dropdowns, with	Since "template" values are so up in the air considering that there is not such thing a template or typical

Module	Issue	Action
	"template" values.	contract, we can't implement this suggestions.
General Tab/Step	Participant suggested making the ENA Sales Contact field a dropdown.	Make it a dropdown.
Home/Search	The participant wasn't sure how to return to the "main" screen. The Search link is not enough.	Change "Search" link to "Home/Search".
Home/Search	Participant commented that the Home/Search screen is very busy.	The design team believes that onscreen, not on paper, it won't look so busy, but we'll watch out for this.
Navigation Links	Both participants took a few seconds to find the New Contract link.	The design team decided the current navigation system should be fine, given that these elements would be more obvious onscreen (than on paper).
New Community Dialog	When adding a community such as "Boston," the participant wasn't sure what to do with the address field.	Make Name field required and change the label to "Community Name."
New Community/New Contact Dialogs	The participant will most always add a new community or contact for the purpose of adding the data to the contract license. Currently, this is a two-step process (add data, move data from Available list to Selected list).	In the New dialogs, add an option that is selected by default to automatically add the new data to the Selected list. While this would be a nice feature, the design team is concerned about scope creep. We will make this an enhancement request.
New Contact Dialog	Since email notification should be or may be an automatic process, the email field should be required.	Make the email field required.
New Contact Dialog	Neither participant was sure what to make of the Admin option.	Add onscreen help text. Also, the Password field will be dropped, as the system will create a unique password for the user to use when he or she logs in to the view-only app; at login, the user can request his or her password (which is emailed to him or her).
New Contact Dialog	The design team realized that there should be a way to prevent duplicate records being created for users already in the Bedrock database.	When the user clicks the Admin option, SLS compares the entered user data with the Bedrock database. If a record is found with a matching email address or a matching last name, the Duplicate Contact Found window appears. The user can select

Module	Issue	Action
		an existing record or click Cancel to create a new record.
New Contact Dialog	The participant question how he could add a new value to the Type dropdown.	Make the dropdown editable.
Search, Advanced	The participant wondered how he could view all contracts expiring in the next month.	Add an option to search "on this date or between these dates" for date fields in the Advanced Search screen.

Test Script Issues

Usability testing revealed the following problems with the test script:

Issue	Action
The participant had several questions about the purpose and context of the application.	Add more information from the business requirements to the introduction portion of the test script.
The design team felt that the "Save & Repeat"/"Save & Add" button labels in the Add Contact and Add Community dialogs weren't really tested.	Add a step to the test script that has the participant add more than one contact and more than one community.
The design team pointed out that we didn't test the Licenses tab.	Add a step to the script to have the user look up how many licenses are in use.
The participant never used the Help button.	Add text to the script saying something like, "Feel free to click the Help button to request more information."
The participant was not sure what to do with dropdown lists.	State at the beginning of the test what dropdowns look like and that when the participant clicks one, he or she will get a list of items or will be told the list of items from which to choose.