MONTHLY NEWSLETTER

September 2009 - Pragmatic Software Newsletters

15 Useful Test Cases for ensuring Consistent User Interfaces

When testing user interfaces, it is easy to overlook test cases that ensure that your user interface is user-friendly and consistent. This newsletter identifies 20 test cases that might be considered when testing user interfaces for consistency.

15 Useful Test Cases for ensuring Consistent User Interfaces

1. **Screen Font Type** - Ensure that the screen font family matches from screen to screen. Mismatching fonts within the same sentence and overuse of different fonts can detract from the professionalism of your software user interface.
2. **Screen Font Sizes** - Ensure that the screen font sizes match from screen to screen. A good user interface will have an accompanying style guide that explicitly defines the font type and size for headers, body text, footers, etc.
3. **Colors** - Ensure that screens do not use different color sets as to cause an inconsistent and poorly thought-out user interface design. Your style guide should define header colors, body background colors, footer colors, etc.
4. **Icons** - Ensure that icons are consistent throughout your application by using a common icon set. For example, a BACK link that contains an icon next to it should not have a different icon on one screen versus another. Avoid free clip-art icons, opt for professionally designed icons that complement the overall look and feel of your screen design.
5. **Narrative Text** - Having narrative text (screen instructions) is a great way to communicate how to use a specific screen. Ensure that narrative text appears at the same location on the screen on all screens.
6. **Brevity** - Ensure that narrative text, error messages and other instructions are presented in laymen's terms but are brief and to-the-point.
7. **Dialog Box Consistency** - Use a style guide to document what choices are available for dialog boxes. You should have not have Save/Cancel dialog on one screen and an OK/Cancel on another, this is inconsistent.
8. **Links** - If your application has links on the screen (e.g. Save as Spreadsheet, Export, Print, Email, etc.), ensure that the links have consistent spacing between them and other links, that the links appear in the same order from screen to screen, and that the color of the links are consistent.
9. **Menus** - If your application has menu items, ensure that menu items that are not applicable for the specific screen are disabled and the order in which each menu item appears is consistent from screen to screen.
10. **Buttons** - If your application has buttons (e.g. Submit, OK, Cancel, etc), ensure that the buttons appear in a consistent order from screen to screen (e.g. Submit then Cancel).
11. **Abbreviation Inconsistencies** - If your screens contain abbreviations (e.g. Nbr for number, Amt for amount, etc), the abbreviations should be consistent for all screens in your application. Again, the style guide is key for ensuring this.
12. **Delete Confirmations** - It is a good practice to ask the user to confirm before deleting an item. Create test cases to ensure that all delete operations require the confirmation. Taking this a step further, it would also be great to allow clients to turn off specific confirmations if they decide to do this.
13. **Save Confirmations** - It is good practice to ask the user to confirm an update if updates are made and they navigate to another item before explicitly saving. Create test

cases to ensure that all record movement operations require the confirmation when updates are made. Taking this a step further, it would also be great to allow clients to turn off specific confirmations if they decide to do this.

14. **Grammar and Spelling** - Ensure that you have test cases that look for grammar or spelling errors.
15. **Shortcuts** - If your application allows short cut keys (like CTRL+S to save), ensure that all screens allow using of the consistent shortcuts.

By the way, did you notice any inconsistency above? We intentionally showed 15 useful test cases above but the narrative before the listing shows 20 test cases. Inconsistencies are easy to spot, right?

Helpful Resources

Below are some helpful resources and templates to aid you in developing software solutions:

- **Software Planner** - http://www.SoftwarePlanner.com
- **AutomatedQA TestComplete (Automated Testing Tool)** - http://www.TestComplete.com
- **STAR QA (Automated Testing Resources)** - http://www.star-qa.com
- **Software Development /QA Templates** - http://www.pragmaticsw.com/Templates.asp
- **Test Case Training** - http://www.PragmaticSW.com/Services.asp
- **Pragmatic Agile Development** - http://www.pragmaticsw.com/PADOverview.pdf

About the Author

Steve Miller is the President of Pragmatic Software (http://www.PragmaticSW.com). With over 24 years of experience, Steve has extensive knowledge in project management, software architecture and test design. Steve publishes a monthly newsletter for companies that design and develop software. You can read other newsletters at http://www.PragmaticSW.com/Newsletters.asp.

Pragmatic Software Company, Inc.
7935 E. Prentice Ave, Suite 105
Greenwood Village, CO 80111 USA
Tel:+1 303.768.7480