

Its Drinking Time!!'s Rebuttal of the Usability Review conducted by Team Goodfellas

07/23/2007

In Team Goodfellas heuristic evaluation of our graphical user interface, three main categories of errors and shortcomings were outlined.

Recognition and Recall

The first concern expressed by Team Goodfellas was that our use of modular dialogs served to break the connection between the system and the real world, by not allowing the user to view related information. However, simply allowing for a more relational view of projects, bugs, and reports, as recommended by Team Goodfellas, does not necessarily rectify the problem of system and real world disconnect. Because of the conceptual encapsulation of bugs being contained within a project and project reports almost exclusively being contained to within a single project, providing a purely relational view of Techzilla, while eliminating the disconnect between the system and the real world, would sever the conceptual connection between the relationship of the system elements. Also, providing a relation view of projects and bugs would avail itself to greater user error and information overload, since clicking on or having the wrong project highlighted in the left pane would greatly effect the display of the bugs in the right pane, while providing little visible feedback as to the current state of the user's selection. As a result, in our refinement of the graphical user interface, we would stick to a linear model of viewing the elements of the system, but would incorporate something similar to breadcrumbing to provide better information to the user in regard to their current position in the system. Also, we would work to make sure that the dialog boxes that appear at each level (system, project, bug, report) would be distinct from to one another, so it would be easier to discern which level in the system you were at by simply looking at the layout of the dialog box.

The second concern expressed by the evaluators was that there was not safe guard against accidentally closing the a dialog window. The evaluators recommended providing pop-up dialogs seeking confirmation to close a window. However, we feel that this solution only leads to the annoyance and belittling of the user. While a settings option allowing for the enabling of confirmation dialogs would help some users, a better solution would be to redesign the closing mechanism so that it was harder to click on accident, thus limiting the number of accidental clicks, while not badgering the user with endless confirmation requests and constantly second guessing the user.

The last two concerns where that it was easy to forget where options were inside the interface and that it was difficult to quickly jump between different levels and objects in the system. In

addressing these final two concerns, we would implement the changes that were laid out in relation to dealing with the first concern, the fact that the system was disconnected from the real world.

Consistency and Standards

The first issue that the evaluation team brought out in regards to the consistency of the application was the use of the term report. In our project, we seemed to use the term report to describe both the submitting of an individual bug report and the viewing of summary data and charts related to a collection of bug reports. The contributing cause of this confusion was the fact that as a team, we had a cloudy understanding of the term as it related to the project specifications. So, the first thing that we would do to rectify the problem would be to make additional inquiries as it relates to the product specifications from the customer in order to answer questions that we had about the application. Second, we would refrain from using the term report when talking about summary charts and figures comparing several bugs, instead referring to them as summary information in general, and more specifically as summaries.

The second issue highlighted also had to do with the word report, this time in dealing with the filing of individual bugs. While the evaluation team recommended that we use button terms such as "submit bug" or "submit defect", we believe that this in itself is confusing, because what you are actually submitting is not a bug, but a report about a possible bug. As a result, we would limit the use of the word report to individual bug report filings, and would thus label the submission button as "Submit Bug Report."

Lastly, the evaluators found our use of a button titled "view users" as our user management dialog to be confusing. In response, they recommended that we instead label the button "User Administration." In correcting this confusion, we would use the label "User Management", since we feel that it is less bulky and more in line with common computer lingo.

User Control and Freedom

The last issues that the evaluation team discussed in their report had to do with the project deletion mechanism. They found problems with the fact that only current projects could be deleted, an error message involved with project deletion was cryptic, and that project deletion did not make use of a confirmation dialog to avoid accidental project deletion. Other than the issue of the cryptic error message, which was actually a typing error, the issue of how to delete project is more complex than following the recommendations made by the evaluators would have one believe. As discussed earlier, our use of a linear instead of overarching relational data structure is meant to limit user error, by in this case, taking the extra step of selecting the project to delete, before being able to actually delete it. Because of the user base, which will be typically programmers, who are advanced computer users, in the redesign of our program, we would shy away from the use of confirmation dialogs,

because they only tend to annoy the advance user. But, in this instance, because of the rarity of deleting projects and the disconnect between deleting projects from the individual, active project's window; we would redesign our application to, in this instance, utilize confirmation windows and allow for projects to be deleted from the top level system view. In regard to the error message, in the reimplementation, not only would the typo be corrected, but it would also be better if we removed the verbiage "our system" from the error message, and left it more generalized as "the system."

OO Design Impact on Reimplementation

Overall, our object oriented design of the application will facilitate the proposed changes to the graphic user interface. Because our design utilizes a facade data structure, the user interface has almost no connection to the back end logic, thus limiting the scope of the changes to the interface code. The only caveat possibly influencing the changes is the fact that our class naming structure does not necessarily match the user interface naming convention. While only confusing to the programmer and not directly related to OO, renaming classes is relatively easy in the VisualWorks environment.