

Program Satisfaction Based on the Perception of Bugs as Features



ABSTRACT:

How much does telling the user that a bug is a feature affect how satisfied they are with using the program? I modified the responsiveness of the UI of ArgoUML and conducted an experiment in which I asked users to make UML class diagrams. The result of this experiment was that you should not tell users about bugs in your program

BACKGROUND:

- Herzig et al. have explored using machine learning to see how misclassifying bugs as features affects bug prediction [1]
 - They concluded that machine learning would predict bugs as features too often and that bug checking should be done by a human
- Gajos et al. have explored how three different types of toolbars in Microsoft Word affected user satisfaction [2]
 - They concluded that the Split UI performed the best, but was not significantly better than the Moving UI

EXPERIMENT DESIGN:

- Modified three aspects of the UI:
 - When hovering over something, it will take 2000ms for the tooltip to pop up.
 - When clicking to make a class object, it will take 1000ms for the rectangle to appear.
 - When clicking to make a class object, the rectangle will be offset by a random amount between -10 and 10 units in the x- and y-directions.
- Three groups of participants:
 - Control**: gave them unmodified ArgoUML.
 - “Bug”**: gave them modified ArgoUML and told them I optimized the UI.
 - “Feature”**: gave them the modified ArgoUML and told them that I edited it.

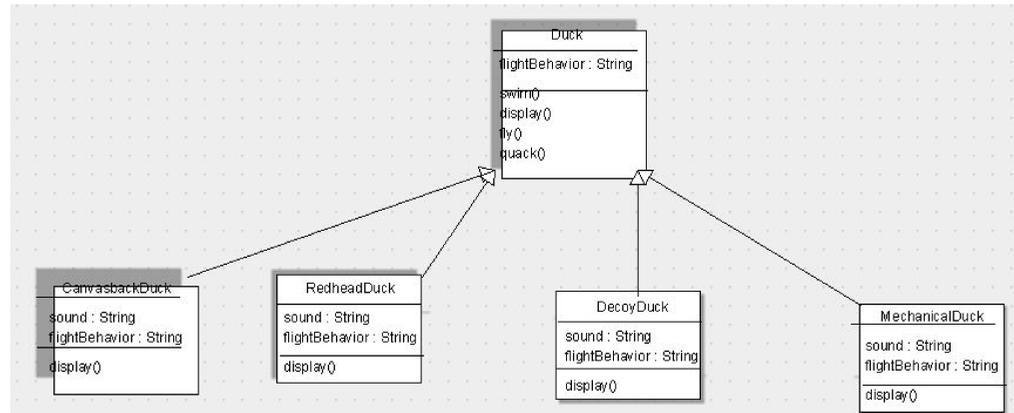


Figure 1. An example of a UML class diagram made using the modified version of ArgoUML.

EXPERIMENT:

- Each participant made three class diagrams of varying complexity
 - Recorded how long it took them to complete each diagram
- Following this, each participant filled out a survey and was asked to rank various aspects of the UI on a scale from 1-5.

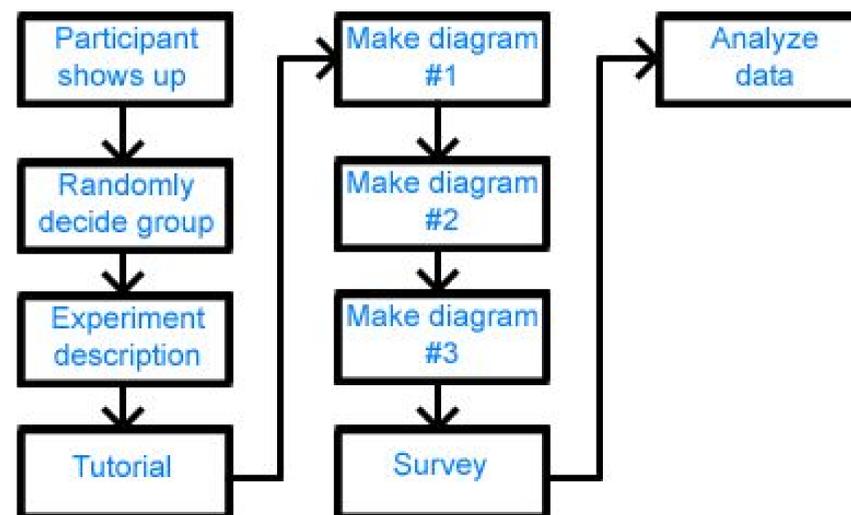


Figure 2. A picture of my experimental procedure.

HYPOTHESES:

- There are two possible hypotheses I have:
 - The “feature” group rates the program as more usable than the “bug” group
 - The “bug” group rates the program as more usable than the “feature” group

DATA ANALYSIS:

I ran two-sample chi-squared tests on the results of the survey and found that there were statistically significant results in four areas:

- Tooltip responsiveness for the control group versus the feature group
- Tooltip responsiveness for the feature group versus the bug group
- Click accuracy for the control group versus the feature group
- Frustration for the control group versus the feature group

I also ran two-sample t-tests on the time data but there was no statistically significant data for these tests.

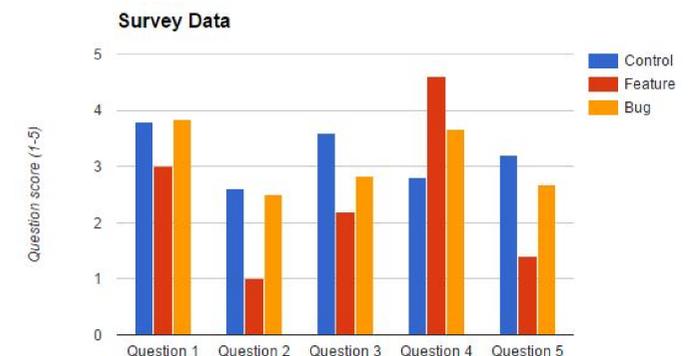


Figure 3. The average response values for each of the five survey questions that I was interested in.

CONCLUSIONS:

- My modifications likely had no effect on how long it takes to draw a UML diagram.
- My data suggests that it might not be a good idea to tell users about bugs in your programs.
 - The feature group reported significantly lower scores on the survey than the control group.

Citations:

- [1] Kim Herzig, Sascha Just, and Andreas Zeller. 2013. It's not a bug, it's a feature: how misclassification impacts bug prediction. In *Proceedings of the 2013 International Conference on Software Engineering (ICSE '13)*. IEEE Press, Piscataway, NJ, USA, 392-401.
- [2] Krzysztof Z. Gajos, Mary Czerwinski, Desney S. Tan, and Daniel S. Weld. 2006. Exploring the design space for adaptive graphical user interfaces. In *Proceedings of the working conference on Advanced visual interfaces (AVI '06)*. ACM, New York, NY, USA, 201-208.