Experiments

Spencer Smith
McMaster University, Canada
smiths@mcmaster.ca

Jacques Carette
McMaster University, Canada
carette@mcmaster.ca

Olu Owojaiye
McMaster University, Canada
owojaiyo@mcmaster.ca

Peter Michalski
McMaster University, Canada
michap@mcmaster.ca

Ao Dong
McMaster University, Canada
donga9@mcmaster.ca

1 User Experiments

1.1 Usability Experimental Procedure

1.2 Procedure

1. Survey participants to collect pre-experiment data
2. Participants perform tasks
3. Observe the study subjects (take notes, record sessions(OBS screen recorder), watch out for body languages and verbal cues)
4. Survey the study subjects to collect feedback (post experiment interview)
5. Prepare experiment report
6. Perform pairwise comparison analysis
7. Prepare analysis report

1.3 Task selection criteria

**The task selection will be determined with the aid of the domain experts attached to any of the selected projects.
**The domain experts will be asked to consider the below criteria when defining a task.
**Domain experts will also be asked to identify what background knowledge is necessary for the suggested tasks - Novice, Intermediate, Advanced

1. Collectively all tasks should not take no more than 2 hours.
2. Selected tasks should reflect common use cases of the software.
3. Include tasks that require a set of sequential or hierarchical steps to be completed

1.4 Usability Questionnaire

Two sources of standardized usability questionnaire we could use.
- https://www.usabilitest.com/sus-pdf-generator- 20-29 - SUS.
- https://uiuxtrend.com/pssuq-post-study-system-usability-questionnaire/ - PSSUQ

2 Modifiability Experiments

[add preamble here describing some of the notes below —PM]

What can be done to measure modifiability?

Idea: measure sensible changes

Can we measure sensible changes? How?

We could gather qualitative data about the process of attempting to make sensible changes. Perceived difficulty, available documentation, prerequisite knowledge, time/effort required, tools required, errors and issues encountered and ability to overcome those issues with the available resources, is there dev support? How to interpret results?

What does a small change mean? What are likely changes - See Parnas

Parnas identifies likely changes as:

Likely Changes: if the system is required to be easy to change, the requirements should contain a definition of the areas that are considered likely to change. You cannot design a system so that everything is equally easy to change. Programmers should not have to decide which changes are most likelyParnas and Clements [1986]

We want to be explicit about what we want to modify.

2.1 Procedure

[Comment from meeting: Instead of identify likely changes, start with Domain Expert about the likely changes they might make (in such a domain?)(write all down), take that list and see which of those likely changes the software is most likely designed for —PM]

1. Identify likely changes using procedure x
2. Select from likely change list (using what criteria)
3. Gather relevant documentation?
4. Ask the domain expert to make the (selected) likely change(s)
5. Have the domain expert answer a questionnaire about their attempt to make the likely change

References

David Lorge Parnas and Paul C Clements. A rational design process: How and why to fake it. *IEEE transactions on software engineering*, (2):251–257, 1986.