Course "Empirical Evaluation in Informatics" SoSe 2015

Freie Universität Berlin, Software Engineering Research Group
Prof. Dr. Lutz Prechelt, Franz Zieris

Practice sheet 6 **Survey: Revise Design** due on 25.05.2015

Tasks **6-1** and **6-2** are a means of quality assurance targeting the high-level design of the survey. Therefore, this practice sheet is about the *idea* of the survey and *not about the details* of its implementation such as graphical layout and formulation of individual questions.

Task 6-1: Peer-review survey design

due on 21.05.2015, 10pm

1. Take a look at the wiki-pages of **at least two** other groups and review their survey design with respect to the following criteria:
   - Is the overall goal of the survey clear?
   - Is the target group described (briefly)?
   - Is the research question clearly formulated?
   - Do the authors explain how they plan to obtain a high number of participants and a good return rate?
   - With regard to the linked first version of their questionnaire: Is it clear how the questions relate and contribute to the stated research question?
2. Please provide *constructive* feedback to the survey's authors via e-mail (you should find their addresses on their project's wiki-page) and add the teaching assistant (zieris@inf.fu-berlin.de) as a CC recipient.

   Remember: You will also receive feedback from your class mates, so aim for a level of clarity and objectivity you would like to find in your own inbox.

Important: Your class mates need your feedback for working on the next task. Please make sure you send it to them *before* Thursday, 21.05.2015, 10pm.

Task 6-2: Revise own survey design

Improve your survey design based on the feedback provided by your class mates. Make sure you understand each point of criticism before duly changing your design. If in doubt, feel free to ask your reviewers.

Task 6-3: Practice R, part 3

In this task, let us revisit the CVS data from sheets 2 and 3. Answer the following questions using short R programs for each of the three projects JUnit, Jikes, and Zile:

1. Counting files:
   - How many different files were changed according to this data, i.e. how many files' names are mentioned in the CVS log excerpts?
   - Which are top 5 of the most often changed files? (`table`, `sort`)
   - Who many files were changed exactly once, twice, three times, five times, and ten times? (`table`, `order` plus clever indexing)
   - How many files were changed by exactly one, two, ..., nine, or ten developers? (`tapply`, `levels`, `length`, `factor`, `table`)

     Try to achieve the same outcome using `sapply` instead of `tapply`.

2. File types:

- Extend your data frame by the column `filetypef` that contains only the file type. You can extract the filetype from the `file` column. (`sub("^.*\\.([a-zA-Z0-9]*)$", "\\1",...)`
- Per type: How many files of that type were changed? (`sort`, `table`)
- Display as a table with one row: What is the mean number of changes for files per type? (`sapply`, `levels`, `mean`, `table`)
- Add three more rows: number of files, minimum number of changes, maximum number of changes.
- Display the table with proper row labels. (`row.names`)
- Sort the table columns (the file types) by the mean number of changes. (`order`)

3. Does the 80:20 rule apply, i.e. do the most-often changed 20% of the files account for 80% of all changes? (`length`, `levels`, `sort`, `quantile`, `cumsum`, `max`)