Analysis Method

This document is intended to support the "Method" section in the paper.

First level of coding

INITIAL CODES

Action: `IC.ACTION`
User's comments: `IC.IN-VIVO`
Problems: `IC.PROBLEM`
Coder's Insight: `IC.INSIGHT`
Event: `IC.EVENT`

DESCRIPTIVE CODES FOR CLASSIFYING BACKGROUND INFORMATION

Programming languages/IDEs: `BG.TOOLS`
Education background: `BG.EDUCATION`
Project details: `BG.PROJECT`

DESCRIPTIVE CODES TO CLASSIFY SUBJECT OF THE DISCOURSE

Domains
- Significance testing: `DOMAIN.DATA-ANALYSIS`
- Machine learning: `DOMAIN.MACHINE-LEARNING`
- Misc.: `DOMAIN.MISC`

Interface
- Console: `UI.CONSOLE`
- Script files: `UI.SCRIPT`
- Notebooks: `UI.NOTEBOOK`
- Other (file browser, presentations, publications, …): `UI.MISC`

PROCESS CODES TO CLASSIFY DATA WORKER'S WORKFLOW

(To further classify `IC.ACTION`)

Exploring alternatives: `WF.EXPLORING-ALTERNATIVES`
 (further evolved to strategies and problems)
Handling error: `WF.HANDLING-ERROR`
Comparing output from alternatives: `WF.COMPARING-OUTPUT`
Switching from exploration to confirmation: `WF.SWITCHING-TO-CONFIRMATION`
Switching modalities: `WF.SWITCHING-MODALITIES`
Confirming the task: `WF.DOCUMENTING-TASK`
Viewing previous scripts: `WF.VIEWING-PREVIOUS-SCRIPTS`
Viewing past code in console: `WF.VIEWING-PAST-COMMANDS.CONSOLE`
Viewing data (visual + text + variables): `WF.DATA`
Executing code: `WF.EXECUTING-CODE`
 (further evolved to selective execution of code and complete execution of code)
Seeking help: `WF.SEEKING-HELP`
Using VCS: `WF.VCS`
Reordering/structuring code: `WF.REORDERING-CODE`

DESCRIPTIVE CODES TO CLASSIFY THE PROBLEMS

User has difficulty finding the previous step: `PR.PREVIOUS-STEP`
Code hoarding: `PR.CODE-HOARDING`
Over execution: `PR.OVER-EXECUTION`
Documentation/Structuring Code: `PR.DOCUMENTATION`
Code Cloning: `PR.CLONING`
Package Dependencies: `PR.PACKAGE-DEP`
Difficulty in Testing Alternatives: `PR.TESTING-ALTERNATIVES`
Stale Data: `PR.STALE-DATA`
"Lost" Code: `PR.LOST-CODE`
Debugging-Related: `PR.DEBUGGING`
Data Dependency: `PR.DATA-DEP`
Task switching: `PR.TASK-SWITCHING`
Missing tooling: `PR.MISSING-TOOL`
Can't find variable: `PR.VARIABLE`

Emotional codes were applied in a focussed manner on particular codes e.g., over execution, code hoarding, and on in-vivo codes generated on a selective basis.

Second level of coding

We generated a code landscape for the first level codes, not unlike the ones above. Then, we used axial coding as the second level of coding. We used the following models to generate the axial codes: action/interaction, causal conditions, and phenomenon.

Some theories are not discussed in the paper because of lack of space/doubts over its validity.

Other details

We used the Coding Manual for Qualitative Researchers as the main reference for coding techniques. Interviews were transcribed manually. We used Evernote to organise the codes (using tags), but also to maintain memos and the codebook.

First 14 participants were recruited through convenience sampling; the last 7 participants through theoretical sampling.

For reporting quotations from our participants in our paper, we used the APA guidelines (https://www.unb.ca/fredericton/studentservices/_resources/pdfs/wss/apaquotations.pdf).