Hypothesis Study Protocol

Before the study session

1. The participant contacts the research for a timeslot
2. The researcher:
    a. Schedules a time and sends the participant an invitation for the agreed time
        i. Remove Google Meet and add the Zoom link on the invite
        ii. Provide a copy of the screener (Please complete this survey)
    b. Participants sheet:
        i. Main tab: Log the planned session date and the researchers assigned
        ii. Xref tab: Fill in name and email beside the next participant id.
    c. GitHub
        i. Go to: https://github.com/cmumatt-org/hypothesis-study-2024
        ii. Click “Use this template,” cmumatt-org / 2024-H##, private, create.
    d. Scoring sheet
        i. Create a participant-specific copy of the Scoring Sheet (H##)
        ii. Fill in the participant id, and date

During the study session

IMPORTANT: Follow the protocol exactly as written, including what to say. Do not ad lib. There should be minimal variation, and only when the participant asks questions, in which case we should give short, pointed answers and avoid any and all discussion.

Note: One or two researchers may be present. If two, here is the work breakdown.

Talker	Scorekeeper (SK)
- Reads what is in the scripts - Watches ensure the participant follows the instructions. - Uploads recordings & repo data - Issues payment to the participant	- Ensures the timers are running - Records the score and start/end times - Tells the talker what the next task is - Updates the Participant sheet

Everyone meets over Zoom at the specified time.

e. Brief "Hi, how are you, thanks for coming. My name is x, and I'm a student researcher at Carnegie Mellon University in Pittsburgh, PA" talk is fine and normal here, but keep it brief and defer questions: there is no time.

SK: Mark Session Happened = X in the Participant sheet

SK: Open the participant's **scoring sheet** - it provides the task sequence

SK: Ensure the clock on your computer is in the format hh:mm:ss (it shows seconds)

Read the consent script:

**** Do not change this IRB-approved text ****

Carnegie Mellon University is conducting this study to evaluate the relative usability of test generation tools used by programmers. The goal is to use the results to design tools that align with programmer workflows and help find bugs more quickly and accurately.

During this study we will ask you to test and debug several programs using different testing tools. Consequently, this study requires experience with programming and testing. These tasks are expected to take less than 90 minutes.

We do not expect that this study will put you at any risk, nor are there any direct benefits to you. For your participation you will receive a $30 Amazon gift card. Participation is voluntary and you can withdraw from the study at any time.

The screen and audio of this session will be recorded and stored securely under an anonymous participant number. We will ask you to turn your camera off before we start recording.

Please note that Carnegie Mellon does not control Zoom's terms and conditions or how Zoom uses any data that Zoom collects from its account holders. A de-identified transcript or quotes from the session may be shared with the research community. Please do not reveal any private or personally-identified information in your responses and please find a private place so we don't accidentally record anyone else.

If you need to contact the researchers at any time please do so by emailing: programming-study-infrastructure@googlegroups.com

Do you confirm that you are at least 18, have had your questions answered, and wish to participate in the research?

If you have questions or concerns about your rights as a research participant, you can contact the Carnegie Mellon IRB at (412) 268-4721, irb-review@andrew.cmu.edu. You will be provided with a copy of this informational script for your records.

SK: Mark Consented = X in the Participants sheet

Read the **Setup Script** below

Thanks!

We will be working in GitHub CodeSpaces for this study. CodeSpaces provides Visual Studio Code and a development environment inside a web browser. To provide you access to the study materials, can you tell me your GitHub id?

[Get the GitHub id, go to https://github.com/cmumatt-org → H## repo, click Settings→Collaboratorators and teams, click **Add people**, enter the user's GitHub id, click **Write** access, then click **Add [GitHub id] to this repository**.]

Ok, can you check your email for an invitation from GitHub to join cmumatt-org? Accepting will give you access to the study repo for the study today.

[Enable participant screen-sharing in Zoom. After the participant accepts the invitation, continue]

Great! Accepting the invitation should have opened the repository for today in your web browser. Can you please share your screen and follow that link? I'll show you how to create a CodeSpace for the study.

[Wait for screen share to come on]

Ok, you're in the **H##** repo. Click the green **Code** button, then the **Codespaces** tab, then click the green **Create Codespace** button. Now the Codespace is starting.

[Wait for the Codespace to be setup and open]

*[**Note**: If using Chrome, they probably need to **NOT** use Incognito mode]*

Great! We're in the Visual Studio Code IDE.

Do you need a minute to set up the IDE the way you usually like it? We just ask that you not install any extensions beyond VS Code's basic Python extensions.

[Wait a minute for them to set it up the way they want]

Ready to begin? Let's turn off our cameras and I'll start recording. Is that ok?

Turn off all cameras, turn on screen share, **start cloud recording**
(You must use cloud recording with transcription enabled - test it prior!!)
Read the **Pre-Task Script** below

Thank you for participating in our study. Today we will be testing several **Python** programs using **Hypothesis**, which you might have used before. The purpose of this study is not to assess your programming or testing skills; rather, we want to understand how testing tools are used.

Each program has a comment that describes the inputs the program allows and its expected behavior. In some cases, inputting allowed values to a program might result in the program behaving contrary to its specification. Your task is to find the inputs, if any, that cause the program to behave contrary to its specification. One example could be, "when I input 'a=0' and any value for 'b' then foo() behaves contrary to its specification." At the end of each task, we will ask you to describe the bugs you found, if any. Do you have any questions?

When you are satisfied you found all the bugs (if any) using the testing tool, let me know. I will ask you to describe them and your level of confidence in the space we provide below the program, save, commit, and push to the repo. Once you move to the next program, you can't go back.

There are no syntax errors in these programs. You don't need to test inputs not allowed by the program.

Before you use the testing tool, we will show a brief tutorial. Please do the tutorial, even if you are familiar with the tool. We will tell you which testing tool to use for each program. Use only that testing tool for that program. Do not try to repair or edit the program.

During the study, we want you to say out loud what you are thinking and doing as you go about your work. If you stop talking, we will remind you to resume by saying something like "please continue speaking."

For each program, we will stop you after thirty minutes, but we won't warn you if you are about to run out of time. It's ok to run out of time. Do you have any questions?

In the root of the repository is a Python cheat sheet in case you need help with Python syntax. You are not allowed to use the internet or take breaks during this study. Please take a moment to put your phone in do not disturb mode. (wait for them to do so) Do you have any questions before we begin?

SK: Start the task sequence in the scoring sheet. You will run two timers on Google:
- 30 minutes task timer (excluding tutorials, which are not timed)

SK: States the first task and tool.

For tutorial tasks

f. **SK**: Ensure the timer is paused.
g. **SK**: Mark the task start time in the scoring sheet
h. Read this script:

> We are now going to do the tutorial for [tool]. This is not timed. Even if you have used this tool before, please follow the tutorial.
>
> Please close all tabs in VS Code
> Expand the **tutorials** folder.
> Right-click on **tutorial-**[tool]**.md** then click **Open Preview**.
> Please follow the tutorial.
> Let me know when you are done.

i. Wait for the participant to finish.
j. **SK**: Mark the task end time in the scoring sheet
k. **SK**: State the next task and tool.

At each task start: (not for tutorials)

l. Read this script:

> For this task, please close all your tabs in Codespaces.
> Open **##/ex##.py** and **##/ex##.py**.
> In the terminal, cd to the **##** directory
>
> We will use Hypothesis for this task. For this task, you are trying to find any allowed inputs under which the program behaves contrary to its specification.
>
> The comment at the top of the program indicates what the allowed inputs are and the program's expected behavior. You can assume only allowed inputs are passed to the program. Start Hypothesis by running **pytest** on the command line. To see print statements, use pytest's **-s** switch.
>
> While you go about your testing, please say out loud what you are thinking and doing. If you stop talking, we will prompt you to resume talking.
>
> When you think you have found all the bugs, or if you think there are none, let me know before you start typing up the report at the bottom.
>
> Do you have any questions before we begin?

m. **SK**: Start the 30 minute task timer.
n. **SK**: Mark the task start time in the scoring sheet.

While each task is running ensure:

o. The participant uses the assigned tool for each task
p. The participant is talking out loud about what they are thinking and doing. If they stop for any reason, prompt them to resume: e.g., "please continue speaking."
q. The participant is not ignoring Hypothesis and only writing PyTest unit tests
r. The participant is not trying to fix or repair the code – stop them if they do this.
    i. Adding print statements or whatever to test is ok.
    ii. After they log their input/output pairs and save, move to the next task.

At task end (participant runs out of time **or** says they are done)

s. **SK**: Stop and reset the 30 minute task timer.
t. **SK**: Mark the task end time in the scoring sheet & move to the next task.
u. **SK**: Update the scoring sheet as the participant is working on all this.
v. Read this script:

> Please take a moment to review the tests you saved as well as the bugs, if any, that you found.
>
> Use the template provided at the bottom of the file to describe the general circumstances, if any, under which the program behaves contrary to its specification. Remember to be specific about the inputs under which this situation occurs, such as "if I call foo(a,b) w/a=1 and b=<any value>, foo() behaves contrary to its specification." You don't need to explain anything else.
>
> [Ensure the participant is recording the inputs and outputs in an unambiguous way. If needed, tell them them to unambiguously state inputs & outputs]
>
> Take a moment to think about how confident you are that you found and reported all the inputs that elicit bugs in the program. Please rate your confidence 1 to 5 where 1 is not at all confident and 5 is very confident.
>
> Please save, close your VS Code tabs, stage, commit, and push to the repo using the user interface. The commit message does not matter: just use something like "update." Use **...->Push** to push to the repo.

w. **SK**: State the next task and tool.

At the end of the timer or when the participant is complete:

x. **SK**: Do not score any remaining, unstarted tasks. Close the **scoring sheet**.
y. Read this script:

> This is the end of the study. Can you please save all files, commit, and push to the repo?
>
> [make sure they do this]
>
> Thanks for spending time with these programs and tools today. We are learning so much from this study about how developers use testing tools.
>
> Please close Visual Studio Code now.
>
> We have a brief survey about the tool you saw today – we would like your candid opinions about your experiences. I'm pasting the URL into the chat.
>
> [Paste the link to the post survey into the chat]
> https://cmu.ca1.qualtrics.com/jfe/form/SV_0p6foaFWdS1ep7w (w/demo q'd)
> https://cmu.ca1.qualtrics.com/jfe/form/SV_1S5ZZvcKMw0FZIi (w/o demo q's)

> Are you able to pull it up? Your participant id is [Hxx].
>
> We'll stay on while you complete the survey in case you have any questions.

z. Wait for them to finish the survey. It's ok to explain a question if needed.

aa. After they submit the post survey, say:

> Great! We're all set. Thanks for filling out the survey. Within two weeks you'll receive the $30 Amazon gift card in the email you provided. If you don't get it for some reason, please email me – my email is on the invitation.
>
> Do you have any questions before we go?

bb. **SK**: Mark Post-Survey = X in the Participants sheet.

cc. It's ok to chit chat a bit if there is time.

dd. Everyone says goodbye

ee. End the Zoom session

Exception Handling:

If the participant loses connectivity or accidentally closes the tab, the following protocol applies:

SK: Pause the timer (task)

SK: Add a comment to the End Time in the **scoring sheet** indicating:

ff. What time the timers were paused

Give the participant up to ten minutes to re-connect (do this no more than twice):

gg. Resume the session

hh. **SK**: Unpause the timer once re-connected.

ii. **SK**: Add the resume time to the End Time comment in the **scoring sheet**

jj. SK: At the end of the study, manually calculate the elapsed time in column F

kk. **SK**: Otherwise, record the time normally. The End Time comment indicates we need to subtract some time in data analysis.

If the participant does not reconnect within ten minutes (or disconnects more than 2x):

ll. End the session.

mm. **SK**: Indicate in the **scoring sheet** comment that the session did not resume.

nn. E-mail the participant:

> Hi,
>
> Sorry we got disconnected from the study. Unfortunately, that happens sometimes. The good news is we are still grateful for your time and within about two weeks you should receive your Amazon gift card via email.
>
> Thanks again for participating in the Usable Testing study!
>
> [name and title]

oo. **SK**: Mark **Post Survey?** and **Data Saved?** = "disconnect" in Participants sheet

pp. **SK**: Mark **Exception** = yes in Participants sheet.

qq. We will still pay the participant

If technical difficulties with the tool prevent completing the study

1. **SK**: Mark **Post Survey?** and **Data Saved?** = “failure” in Participants sheet
2. **SK**: Mark **Exception** = yes in Participants sheet.
3. We will still pay the participant

After the Study Session

1. If not disconnected:
    a. Upload the Zoom recording & transcript to the NaNofuzz data folder. Name each file by participant id.
    b. In the Participant sheet, mark Data Saved = X
2. Go to: https://github.com/cmumatt-org → People→Outside Collaborators
3. Click the gear beside the participant → Remove from all repositories
4. Click **Remove outside collaborators**
5. E-mail a copy of the consent script to the participant.
6. In the Participant sheet, mark **Sent Consent** = X
7. Issue payment to the participant (Josh can help with this)
8. In the Participant sheet, mark **Payment Made** = X