

## Interview Questions:

In total, 13 participants

## Demography questions:

1. How many years of experience do you have in total with software development?

<1 year: 0 (number of participants answered this.)

1-2 years: 7

2-5 years: 5

5-10 years: 1

## Questions regarding commit message quality:

Given the definition that a commit message with “Good” quality is the message that contains a summary of the changes in the commit (“What” information) and a description of the reasons for the changes (“Why” information):

2. Do you pay attention to writing a “Good” commit message every time you commit changes to a codebase? Out of writing “What” and “Why”, which one do you pay more attention to? Or are they equally important when you write commit messages?

Yes, pay more attention to “What”: 9

Yes, equally important: 3

Yes, pay more attention to “Why”: 1

3. Do you think the quality of existing commit messages written by others has any impact on the commit that you are about to submit?

Yes: 11

No: 2

4. What effects do you think will have on future software development if you write a “Bad” commit message? For example, a commit message without “What”, “Why”, or both.

Some people gave multiple effects.

It will affect project handover in the future: 1

Hard to track software evolution: 2

Reduce readability of the code base: 1

Hard to understand code changes: 7

Cause future maintenance issues: 5

5. Do you think commit message quality in general is important in preventing bugs and maintenance issues from being introduced to the code base?

It's important in preventing maintenance issues but not very important in preventing bugs: 2

Yes, both: 8

No: 3

6. Do you think it's a good practice to attach (issue/pull request) links instead of directly writing texts in the commit messages to include "Why" information?

Yes: 5

No: 5

Depends on the change size/company: 2

No idea: 1

7. Are you aware of any guidelines about writing commit messages of high quality?

Yes: 10

No: 3

8. What other aspects do you think are also important to writing a high-quality commit message, in addition to it containing "What" (a summary of the changes in the commit) and "Why" (a description of the reasons for the changes)?

Some people gave multiple aspects.

Restricted number of words/ concise/ to the point: 3

A standard writing pattern: 2

Links to other related systems (code review ID, bug tracking ID, discussion...): 5

No idea: 2

Easy to read/ understand: 3

Corresponding test cases information: 2

## Questions regarding personal development in writing commit messages:

In this section, I'd like you to select one choice from the options below that best describes your personal change in writing commit messages over time:

9. What's the best way to describe the evolution of you writing commit messages over time?
- a. I tend to write more "Good" commit messages over time
  - b. I tend to ignore "What" information in my commit messages
  - c. I tend to ignore "Why" information in my commit messages
  - d. I tend to write "What" information in my commit messages
  - e. I tend to write "Why" information in my commit messages
  - f. I am not sure how the quality of my commit messages changes over time.
  - g. I tend to write commit messages that have neither "What" nor "Why" over time/ I tend to not write commit messages at all
  - h. I tend to write messages in some other ways, such as...

If "f" is your choice, what do you think could be the reasons that you're not sure about the change:

- 1. I didn't pay attention to the quality of commit messages I wrote
- 2. I followed guidelines/others' commit messages writing styles when I make my commits, so I don't know
- 3. Other reasons?

If one of a-e or g is your choice, what do you think could be the reasons?

A: 7

Save people's time, make code base clean, improve working efficiency, big companies' guideline

B: 0

C: 1

adding why makes commit messages unnecessarily long,

D: 4

Use links to provide Why, team's culture,

E: 0

F: 1

Follow other's writing.

G:0  
H:0