Study on annotating implementation and design choices

*Required

Annotations Protocol Investigation

1. How frequently do you comment your source code (or make annotations elsewhere) for documenting your design and implementation choices? *

   Mark only one oval.

   ○ Never

   ○ Less than 25% of my development tasks

   ○ Between 25% and 75% of my development tasks

   ○ More than 75% of my development tasks

2. How frequently do you insert comments for annotating delayed work activities, temporary patches, or, in general, source code not ready yet (e.g., TODO, FIXME, hack, workaround, etc)? *

   Mark only one oval.

   ○ Never

   ○ Less than 25% of my development tasks

   ○ Between 25% and 75% of my development tasks

   ○ More than 75% of my development tasks

3. Does any of the projects for which you contribute have specific policies regarding the way design and implementation choices should be documented (e.g., in the source code)? If yes, please describe them. *

4. Does any of the projects for which you contribute have specific policies for annotating delayed work activities, temporary patches, or, in general, source code not ready yet (e.g., TODO, FIXME, hack, workaround, etc)? If yes, please describe them. *

5. What are the tracing mechanism mainly used for annotating your implementation and design choices? *

Tick all that apply.

□ Source Code
□ Task Management Tools (e.g., Trello)
□ Internal Mailing List
□ Messaging Apps (e.g., Slack)
□ Code Reviews or Pull Requests
□ Issue Trackers
□ Commit Messages
□ Internal/Private Documents/Sheets

Other: □ ____________________

6. While annotating your implementation and design choices in the source code do you rely on IDE-supported annotations (e.g., automatically generated TODO)? *

Mark only one oval.

○ I don't use an IDE/my editor does not support automated annotations

○ No, I do not use annotations at all

○ No, I ignore the IDE-supported annotations

○ Yes, and I will not spend time in modifying their content

○ Yes, but I will spend time modifying their content addign more useful information

○ Other: ____________________________

7. What are the main motivations for annotating the implementation and design choices? *

Tick all that apply.

□ Reminder for yourself

□ Reminder for the community, the development team, or the whole organization who could understand and improve the source code later on

□ Reminder for newcomers joining/contributing to the project

□ Annotating the presence of a misbehavior, to make everybody aware that the implementation and/or the design is not in the right shape as it should be

Other: □ ____________________________

8. What are the main motivations for NOT annotating your implementation and design choices? *

Tick all that apply.

□ Lack of time

□ I remember perfectly whether any portion of my source code needs to be improved

□ I am shy reporting that my code is not in the right shape

Other: □ ____________________________

9. What is the typical content that you usually include while annotating delayed work activities, temporary patches, or, in general, source code not ready yet (e.g., TODO, FIXME, hack, workaround, etc)? *

Tick all that apply.

□ Just a tag (e.g., TODO, FIXME) as a reminder

□ A sentence explaining that the source code needs to be refactored or, in general, re-designed to make it more maintainable

□ A sentence explaining that the source code needs to be refactored or, in general, re-designed to improve performance

□ A sentence explaining that the program might not properly behave under certain conditions

□ A sentence explaining that we are using a sub-optimal API and a better one needs to be found (you are not aware of that or simply it is not available yet)

□ A sentence explaining that the feature being implemented is not complete

□ A sentence highlighting that your code is a temporary fix that will be modified after a specific event will occur (e.g., fix a bug, release of a new API, implementation of a feature elsewhere in the project)

□ A sentence highlighting the presence of a BUG in the source code

Other: □ ______________________________

10. If you find an annotation reporting that the code is not in the right shape, in the code while implementing a new feature or improving an existing feature, what do you actually do? *

Tick all that apply.

□ If I am the author of the admission, it simply reminds me that I have to find a proper solution

□ If I am not the author of the annotation, I do not try to find a solution for it, but I will consider it while modifying the source code

□ If I am not the author of the annotation, I try to find a solution for the problem stated in the annotation before applying further changes to the source code

□ If I am not the author of the annotation, I simply ignore its content

Other: □ ______________________________

Demographics Information

11. What is your highest education qualification?

Mark only one oval.

○ Ph.D.

○ Master Degree

○ Bachelor Degree

○ Other: ______________________________

12. How many years of development experience do you have?

Mark only one oval.

○ Less than 5

○ Between 5 and 10

○ More than 10

13. How many open source projects did you contribute so far?

Mark only one oval.

○ One

○ Between one and five

○ Between five and ten

○ More than ten

14. What programming languages do you typically use in your development activities?

Tick all that apply.

□ C/C++
□ C#
□ Java
□ Javascript
□ Python
□ Ruby
□ PHP
Other: □ ______________________

15. Which IDEs/editor do you usually use to develop code?

Tick all that apply.

□ No specific IDE/Editor
□ Dev-C++
□ IntelliJ IDEA
□ Eclipse
□ Emacs
□ NetBeans
□ PyCharm
□ Vi
□ Visual Studio .NET
□ Xcode
Other: □ ______________________

16. What Kind of communication channels/Software Configuration Management tools do you typically use?

Tick all that apply.

□ Issues Trackers
□ Pull Requests/Code Review Tools
□ Mailing List
□ SCRUMs board
□ Chats and Other Communication Channels (e.g., Slack or IRC)
Other: □ ______________________