ENPM809W – Introduction to Secure Coding

Homework – 6

Author – Syed Mohammad Ibrahim

UMD ID: iamibi

Email: iamibi@umd.edu

1. How should the proposed checklist for a code review be used by developers and reviewers?

A. The proposed checklist is iterative, that is, it needs to be looked back at by the developers and reviewers as and when they progress with the code keeping in mind about the context of the code changes pushed to a pull request.

2. The checklist references a PR or Pull Request to merge features into the main code base. What does the size and shape of a PR refer to?

A. Having the context of the respective PR's will help the reviewers to make a necessary judgement on the size of the PR and suggest whether the PR is too big and should be broken down into manageable PRs or is small enough to have it reviewed.

3. What does the context for a PR refer to?

A. The context for a PR should answer the questions like what is the PR code changes trying to achieve? or why is it doing these changes in the first place? This should ideally be outlined in the PR top message with a clear context and easy-to-understand language.

4. What should developers include for communicating the relevance of a PR?

A. The developers should be including the complete context and the process they went through to get to these resulting code changes. This provides relevance to the people who are reviewing the code and can better understand the changes.

5. For readability, what are the esoteric language features being referenced?

A. The esoteric features of a language can make a trade-off with readability with little to contribute to the performance. What this essentially does is reduces readability for someone who will work on this code in future, and it will be difficult to backtrack the changes that were intended while writing the code. Thus, writing a naïve readable code which accomplishes the same thing with little performance trade-off is fine if it accomplishes the task.

6. What are some of the rules for naming in a PR? Please explain why each rule is important.

A. Few rules mentioned by the author of the page are:

- Misleading, confusing, or single/double letter variable names should be avoided.

- Unidiomatic names should be avoided for any method/variable as they introduce noise for the person reviewing the code.

- Adding a datatype to the variable name causes information leak and thus, increases the cognitive overhead required to review the code.

7. Gotchas on how code can break things or issues it could raise should be specified as part of a PR. True/False.

A. True

8. It is generally always required to include a statement about security in the PR regarding a change to the code. True/False. Why is this statement important?

A. True, it is always advisable to add any security implications that the code change might be affecting. This will make sure that the code reviewer knows and advises accordingly.

9. A Pull Request as per the author should be descriptive of the changes being made, what it affects, why anyone should care and how it addresses any issues. True or False.

A. True

10. What would you recommend including in a checklist for code review that were not already included in the given list on the author's Github site given above?

A. I would suggest adding the original approach and final approach and how it was bottled down to. This should be included because it helps the reviewer to identify why the final approach is the most optimal solution for the given PR and how much apt it is to the given feature development.