Google

Providing Context with Commit Messages and Bug Reports

You are caught in a trap. Mechanisms whirl around you, but they make no sense. You desperately search the room and find the builder's original plans! The description of the work order that implemented the trap reads, *"Miscellaneous fixes."* Oh dear.

Reading other engineers' code can sometimes feel like an archeology expedition, full of weird and wonderful statements that are hard to decipher. **Code is always written with a purpose, but sometimes that purpose is not clear in the code itself.** You can address this knowledge gap by documenting the context that explains why a change was needed. Code comments provide context, but comments alone sometimes can't provide enough.

There are two key ways to indicate context:

Commit Messages
○ **A commit message is one of the easiest, most discoverable** means of providing context. When you encounter lines of code that may be unclear, checking the commit message which introduced the code is a great way to gain more insight into what the code is meant to do.
○ **Write the first line of the commit message so it stands alone**, as tools like GitHub will display this line in commit listing pages. Stand-alone first lines allow you to **skim through code history much faster**, quickly building up your understanding of how a source file evolved over time.

> **Example:**
> ```
> Add Frobber to the list of available widgets.
>
> This allows consumers to easily discover the new Frobber widget and add it to their
>     application.
> ```

Bug Reports
○ **You can use a bug report to track the entire story of a bug/feature/refactoring,** adding context such as the original problem description, the design discussions between the team, and the commits that are used to solve the problem. This lets you easily see all related commits in one place, and allows others to easily keep track of the status of a particular problem.
○ **Most commits should reference a bug report**. Standalone commits (e.g. one-time cleanups or other small unplanned changes) don't need their own bug report, though, since they often contain all their context within the description and the source changes.

Informative commit messages and bug reports go hand-in-hand, providing context from different perspectives. Keep in mind that such **context can be useful even to yourself**, providing an easy reminder about the work you did last week, last quarter, or even last year. Future you will thank past you!

More information, discussion, and archives:
testing.googleblog.com