Method comments

In general, we do not comment our methods. Beck teaches that a comment is used to indicate that a method is not yet finished. That's how we use them.

When we have a method that is unclear on the face of it, we try to rename or refactor the method so that it becomes clear. If we fail to make it clear, we'll put a comment in it.

If a method uses an unusual algorithm, or one that is published somewhere, to which we wish to refer, we might leave a comment in the method.

If a method has been shown by a performance measurement to be a bottleneck, and we have written it in a peculiar way for efficiency, we will leave a comment in the method.

Probably less than one percent of the methods in the system have comments. If they needed a comment, they would have them. Most methods written using our style do not need them.

On the contrary, all methods should have comments to make the code more clear.

- Properly written Smalltalk code usually expresses intention directly, so that comments are not needed. If code needs commenting, it needs refactoring even more.
- If you have an example of commented Smalltalk code that you think needs its comments, please send it to me and I will see whether I can refactor it not to need them. I'll post the results, either way, on my page. Be sure to include the unit tests!

[Home] [XP] [XP Practices Frame]

ronjeffries@acm.org
http://www.armaties.com