

One Page of Quality

Prof Bill - Oct 2016

Quality only ranks a single page of thought?!? Yup. And I'm proud of it. We are lean and mean in 694. For quality we will (and must!) rely on 1) personal responsibility, 2) team unity, and 3) class reviews of all code and tests.

We have another huge advantage: Ruby on Rails. We are working at a higher level of abstraction in Rails, than in, say, a Java program. We will rely on code (gems) used by thousands of other developers. Finally, testing is part of the very foundation of Rails.

Ruby on Rails Code

A simple rule to follow: Your code should be at least as good as the tutorial.

Here's are my minimal guidelines:

- **Comments** - Quality comments are absolutely necessary for quality code. Don't forget to comment your sources for borrowed code/ideas. Also, I am a BIG fan of a comment header identifying the author and other major contributors.
- **Idiom** - You should follow Rails idioms, ala our tutorial. For example, if you're injecting arbitrary SQL into a method, we need to talk.
- **Indent** - Use consistent indentation in your code. This is especially important in complex HTML/ERB files! 2 spaces for Ruby code, eh.
- **Names** - Use good names for methods, variables, etc. Rely on idiom and approaches used in the tutorial.

Testing

This is more difficult. Maybe not: Your tests should be at least as good as the tutorial. Idea: Each person adds at least three test methods at the end of each sprint.

Sprints and Reviews

There will be code and test reviews at the end of each sprint:

- End of Sprint #1 - Prof Bill will review all code and tests
- End of Sprint #2 - We'll review all code and tests up on the projector in class
- End of Sprint #3 - TBD

thanks... yow, bill