The Ten Commandments of Egoless Programming

The Ten Commandments of Egoless Programming, as originally established in Jerry Weinberg's book The Psychology of Computer Programming:

1. Understand and accept that you will make mistakes. The point is to find them early, before they make it into production. Fortunately, except for the few of us developing rocket guidance software at JPL, mistakes are rarely fatal in our industry, so we can, and should, learn, laugh, and move on.

2. You are not your code. Remember that the entire point of a review is to find problems, and problems will be found. Don't take it personally when one is uncovered.

3. No matter how much "karate" you know, someone else will always know more. Such an individual can teach you some new moves if you ask. Seek and accept input from others, especially when you think it's not needed.

4. Don't rewrite code without consultation. There's a fine line between "fixing code" and "rewriting code." Know the difference, and pursue stylistic changes within the framework of a code review, not as a lone enforcer.

5. Treat people who know less than you with respect, deference, and patience. Nontechnical people who deal with developers on a regular basis almost universally hold the opinion that we are prima donnas at best and crybabies at worst. Don't reinforce this stereotype with anger and impatience.

6. The only constant in the world is change. Be open to it and accept it with a smile. Look at each change to your requirements, platform, or tool as a new challenge, not as some serious inconvenience to be fought.

7. The only true authority stems from knowledge, not from position. Knowledge engenders authority, and authority engenders respect – so if you want respect in an egoless environment, cultivate knowledge.

8. Fight for what you believe, but gracefully accept defeat. Understand that sometimes your ideas will be overruled. Even if you do turn out to be right, don't take revenge or say, "I told you so" more than a few times at most, and don't make your dearly departed idea a martyr or rallying cry.

9. Don't be "the guy in the room." Don't be the guy coding in the dark office emerging only to buy cola. The guy in the room is out of touch, out of sight, and out of control and has no place in an open, collaborative environment.

10. Critique code instead of people – be kind to the coder, not to the code. As much as possible, make all of your comments positive and oriented to improving the code. Relate comments to local standards, program specs, increased performance, etc.