Documentation / comments

Write your pseduocode in English. You have to turn your pseudocode into actual code. Pseduocode is writing your algorithm with no frills.

CRC - Class Responsibility Collaborator.

Refactoring - Making local code simpler.

Why are CRC Cards important? They lay out the static architecture. You can simulate the architecture.

How does the CRC simulation? CRC Cards which is handed out to people and we do a kickoff. One of the card is the user.

READ ME

simple set of instrcutiosn that explain how to get started and what it does.

AUTHORS

CLASS

PARTS/MODULES

How do they come together

How to install

How to run

CODING STYLE

Use google's coding style

put spaces around + sign

If it's easier to read;it's easier to understand.

How Software should be designed

Reasonable person standard. Submit what you have on time and be honest. Ask for help early. Have good social skills.
Smart person standard. Always be ready to show your boss visible progress on what you're working on.

Maintain your morale

If you feel sad or depressed you software making ability will not be as sharp. Try to focus on what you can do in the next hour or in the next 5 minutes.What can you do to make a little progress. A series of success will build your morale, arrange for yourself to succeed.

MINI SWE RULES

0. Fast rule

Get to the beginning, have tangible work. Go ugly early. Clean it up when it works

1. Don't pre-optimize

Premature optimization is the root of all evil. Focus on the now, don't waste your effort now for a reward that might not ever come. Not every part of your code is a bottleneck won't know until it is completed. Profile your code. Find where the bottleneck is and resolve it

2. Kill bug hunts

90% of typical software developing is spent killing run-time bugs. Compilers bugs are quick to fix so make your runtime bugs look like compile time bugs.

3. The 2-story story

Top-story is written in the user's problem domain language
Bot-story is Comp-Sci stuff
User-Scenario: 1 user role; rambling paragraphs on what user does in that role
Use-Case: one role-task; ID agents, actions, interaction messages
write CRC card per agent: class=agent;Responsibilties=actions;collaborators=msg-buddies
simulate CRC architecture: one card per person, sim the task kickoff, interactions, and results

4. Use sample EIO before your code

write expected input-output pairs to drive your detailed design; based on CRC sim architecture.

5. Publish (half-day) goal for your planned day's effort

gives you 100% overrun leeway
gives you massive estimation practice

6. clean the page-before you let your code become visible to others

don't get a reputation for turning in poor quality code
avoid lehman's mummy laws of S/W development life cycle atrophy