Tips And Checklists From The Pragmatic Programmer

Tips And Checklists From The Pragmatic Programmer



- [ ] Care About Your Craft
- [ ] Think! About Your Work
- [ ] Provide Options, Don't Make Lame Excuses
- [ ] Don 't Live with Broken Windows
- [ ] Be a Catalyst for Change
- [ ] Remember the Big Picture
- [ ] Make Quality a Requirements Issue
- [ ] Invest Regularly in Your Knowledge Portfolio
- [ ] Critically Analyze What You Read and Hear
- [ ] It's Both What You Say and the Way You Say It
- [ ] DRY—Don 't Repeat Yourself
- [ ] Make It Easy to Reuse
- [ ] Eleminate Effects Between Unrelated Things
- [ ] Use Tracer Bullets to Find the Target
- [ ] Prototype to Learn
- [ ] Program Close to the Problem Domain
- [ ] Configure, Don't Integrate

- [ ] Put Abstractions in code, details in metadata
- [ ] Always Design for Concurrency
- [ ] Don't Program by Coincidence
- [ ] Estimate the Order of Your Algorithm
- [ ] Fix the Problem, Not the Blame
- [ ] Don't Assume It—Prove It
- [ ] Write Code That Writes Code
- [ ] Use Assertions to Prevent the Impossible
- [ ] Minimize Coupling Between Modules
- [ ] Design to Test
- [ ] Don't Be a Slave to Formal Methods.
- [ ] Test Early. Test Often. Test Automatically
- [ ] Sign Your Work

Make and Share Free Checklists
checkli.com