Basic and Power Tools

*(Pragmatic Programmer, Chapter 3)*

- **Choose one editor and know it well**
  
  You need to be proficient. Simply typing linearly and using a mouse to cut and paste is not enough. You just can't be as effective that way as you can with a powerful editor under your fingers.

- **Configurable, Extensible, Programmable**

- **Syntax highlighting, auto-completion, boilerplate, auto-indent, IDE-like features (compile, run)**

  Using notepad is like using a teaspoon as a shovel—simply typing and using basic mouse-based cut and paste is not enough.
Use Source Code Control

- Good user interfaces have an UNDO feature, preferably multiple levels of undo, redo
  - How do we get this with our programming efforts?
  - Source control, configuration management

- Once a program is released, what's next?
  - Work on next version
  - Work on bug fixes
  - How to do these simultaneously? Branch source tree

- How does more than one person work on same program?
  - **CVS**, concurrent versions system
Debugging

- **Origin of word bug**
  - That's not a bug, that's a feature
  - Call it an error, a mistake, a fundamental flaw, ...

- **Debugging as problem-solving**
  - Don't cast blame, find and fix the bug
  - Resist myopia: don't eliminate symptoms, fix the bug

- **What code to fix?**
  - Compile a clean version, how?

- **How to fix? Use debugger, code walk-through, eyeballs**
  - Read documentation (select system call is broken)
Great Programming

- Everyone else does bad programming
- Everyone else does good programming
- Everyone does mediocre programming
- What is great programming?
- What is bad programming?