The Clean Coder

Chapter 6 – Practicing

Background on Practicing

All professionals practice their art by engaging in skill-sharpening exercises

[Robert C. Martin]

Background on Practicing (con't)

Practicing is not a new concept in software development

Hello World

- Proof we can write and execute any program

In the past: Programmers did not practice *FORMALLY*

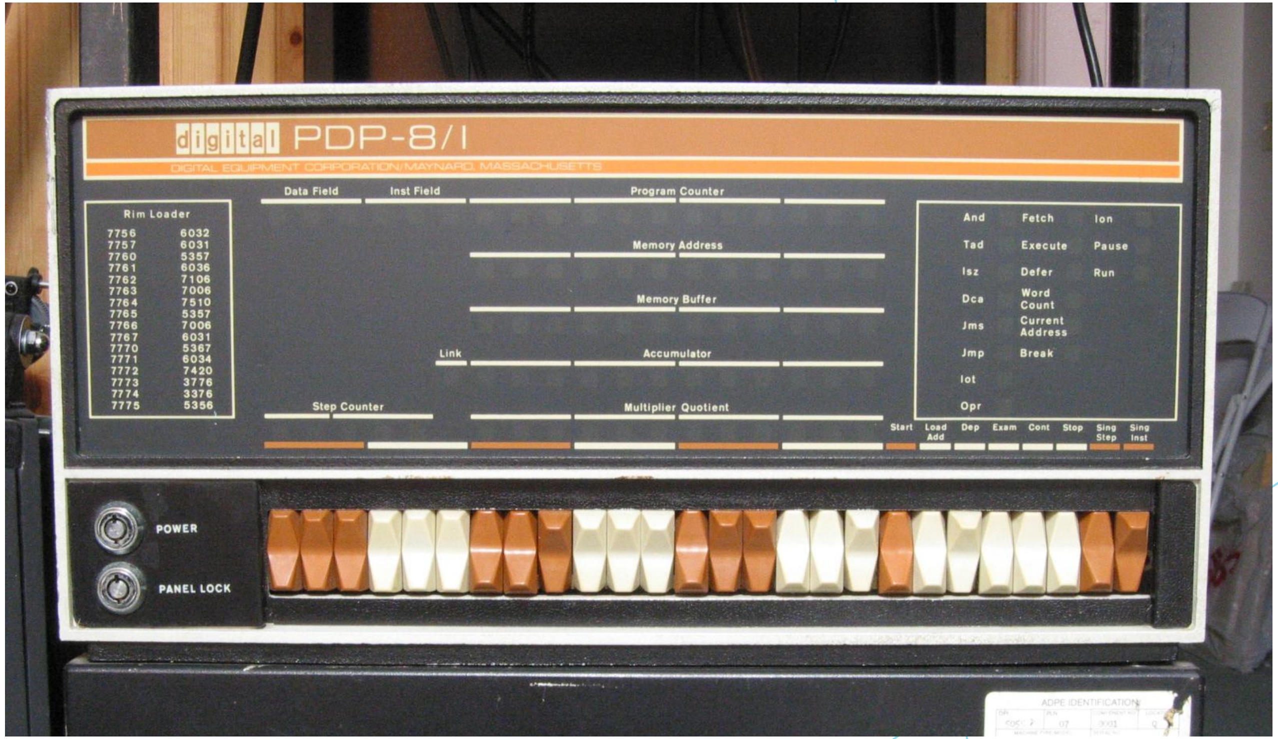
- The overhead involved in writing code was too substantial
- Physical dexterity, deftness or mastery was not required
- Correctness and completeness were the goals

Twenty Two Zeros*

Moore’s Law [1965]

Modern Laptops are 10^22 times more powerful than PDP-8/I (not actually true, but close enough)

We’re doing the same thing as back then but much faster turnaround



Rim Loader

7756	6032
7757	6031
7760	5357
7761	6036
7762	7106
7763	7006
7764	7510
7765	5357
7766	7006
7767	6031
7770	5367
7771	6034
7772	7420
7773	3776
7774	3376
7775	5356

Turnaround Time

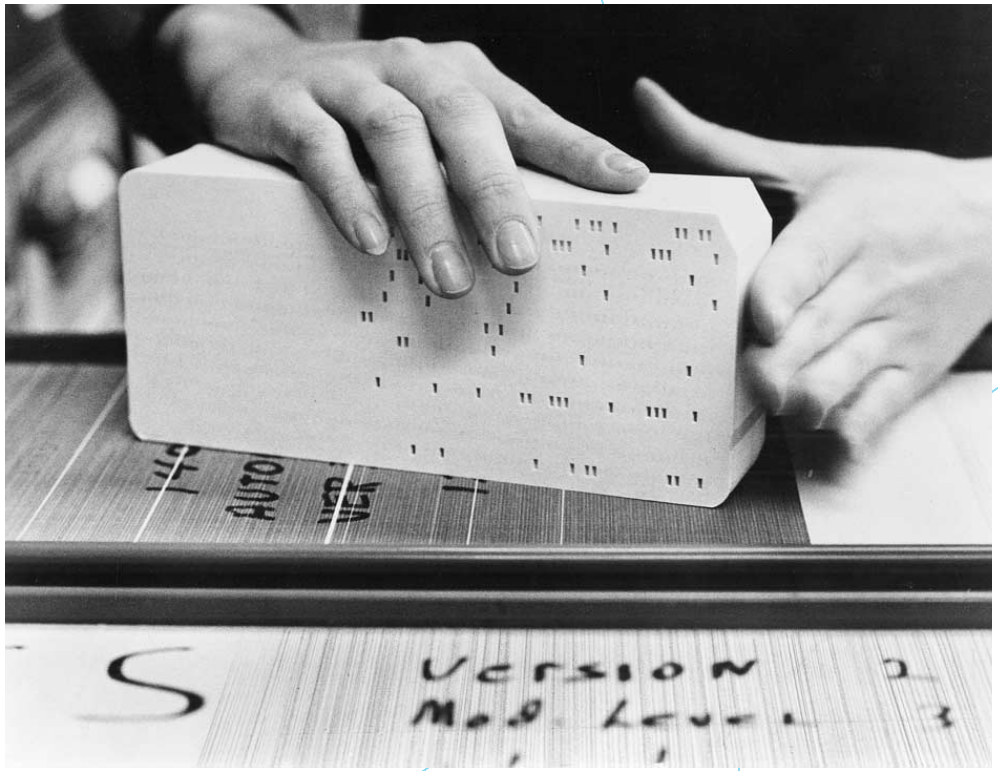
The *WAY* we work has changed dramatically

Compile Times

- 60’s : A day or 2
- 70’s: 45 Mins
- 90’s: Minutes
- Now: Virtually No wait

64,000-line Java Project

- Full build, including all unit and integration testing in 4 Mins
- The Whole QA Process in less than 4 mins



Turnaround Time (con't)

So we have a new priority as coders:

- Code quickly and deftly

Doing anything quickly requires practice

The true value of practice!

- In order to make fast decisions we must “know” exactly what to do

Basically, reacting without thinking

The Coding Dojo

The Bowling Game

- A demonstration of TDD

A Social Space to Practice

- Like a martial arts Dojo
- An informal gathering of programmers working on Katas

Kata

In Martial Arts

A set of choreographed movements that simulates one side of combat

The goal: perfection

The purpose: Train mind and body how to react in a particular combat situation

In Programming

A precise set of choreographed keystrokes and mouse movements

Simulates solving a specific programming problem

One is practicing the movements and decisions involved in solving the problem

Kata (con't)

What are they good for?

- Learning hot keys and navigation
- Learning disciplines like TDD and CI
- Driving common problem/solution pairs into your subconscious

Kata Resources

- http://codekata.com
- https://github.com/gamontal/awesome-katas

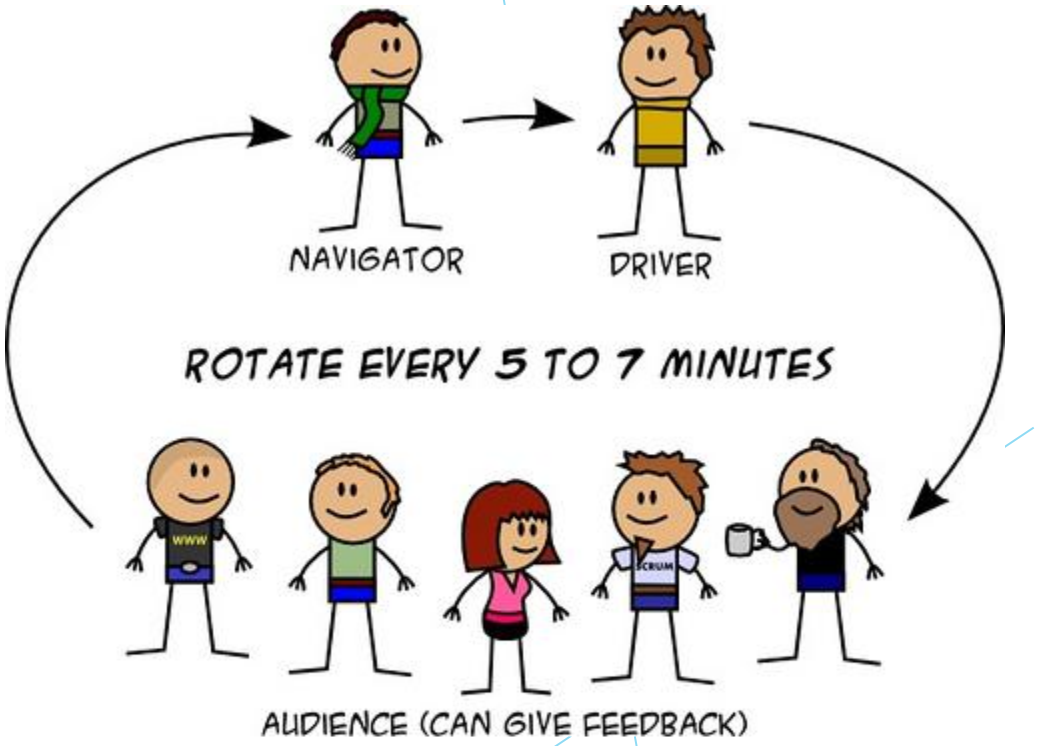
Kata Variations

Wasa

- Two Person Kata
- One programmer write a unit test
- The other writes code to make it pass
- Then they reverse roles and continue

Randori

- Free form “combat” in martial arts
- A game played at many coding dojos
- Like Wasa but in a loop



Broadening Your Experience

- Open Source
  - Pro-bono work like Doctors and Lawyers
  - Work on something outside of your standard skillset
- Practicing Ethics
  - Practice on your own time
  - For yourself: try to practice things you don’t do at work
- Make yourself more prepared and competitive

Resources for Practice

www.repl.it – Compilers and Terminal in Browser

www.codepen.io – Front End Development Environment

Places to Practice

https://www.hackerrank.com/ - General Problems in Many Languages

https://projecteuler.net/ - Math-y problems in CS

https://www.kaggle.com/ - For Data Science