Notes from the Machine Room

by Tomas Brandt

Volume 1

Contents

1 On Reading Code **5**

1.1 Naming Things 5

1.2 Functions Should Do One Thing 6

2 On Writing Tests **7**

2.1 The Cost of a Missing Test 7

Bibliography **8**

Index **9**

Chapter 1

On Reading Code

> "Programs must be written for people to read, and only incidentally for machines to execute."
>
> —Harold Abelson abelson1996structure

Most of the time a working programmer spends is not spent writing new code; it is spent reading code someone else wrote, including an earlier version of themselves. This chapter is about the habits that make reading code faster, borrowing heavily from the discipline described by Martin in Clean Code: A Handbook of Agile Software Craftsmanship (2008).

1.1 Naming Things

A variable name is a small act of documentation. `d` tells the reader nothing; `elapsedTimeInDays` tells the reader everything they need to know without opening the definition. The cost of a longer name is paid once, at the keyboard; the cost of a short, meaningless name is paid every time someone reads the line afterward.

1.2 Functions Should Do One Thing

A function that does one thing is easy to name, easy to test, and easy to reuse. A function that does three things is easy to write and hard to do anything else with. The discipline is not free — it usually means more, smaller functions — but the alternative is a function whose name is a lie the moment it grows a second responsibility.

Chapter 2

On Writing Tests

A test is a specification that happens to be executable. Writing the test before the code forces the question "what should this actually do?" before the more comfortable question of "how do I make this compile?"

2.1 The Cost of a Missing Test

Untested code is not free of bugs; it is merely free of known bugs. The distinction matters because the bugs an untested codebase does have tend to surface in production, in front of a user, at the worst possible time — precisely the outcome tests are meant to move earlier and make cheaper.

Bibliography

[1] Robert C. Martin.
Clean Code: A Handbook of Agile Software Craftsmanship.
Prentice Hall, 2008.

Index

abelson1996structure, Harold Abelson, 5

functions
    single responsibility, 6

Martin, Robert C., 5

naming, 5

tests, 7
    cost of skipping, 7