Debugging is a code smell

Nelson Jovel

Section 1

Introduction

Motivation

What debugger do you use?

"I just use print statements" -Dave Kimura

Scope

Will not include:

- Debugging production errors

Will include:

- Debugging development errors

Section 2

Who am I?

MCIEA


mciea
School Quality Measures Dashboard
School Quality Measures Dashboard
A school quality framework with multiple measures that offers a fair and comprehensive picture of school performance
Select a district
Select a school
Go
Teachers + Leadership
School Culture
Academic Learning
Community + Wellbeing
Resources
School Quality Measures Framework
Through a strand of work led by professor Jack Schneider at the University of Massachusetts Lowell and a team of researchers, MCIEA is reimagining how we measure the quality of our schools and the learning experiences of our students.
To achieve this goal, MCIEA sought input from stakeholders in each consortium district to build a school quality framework that reflects what the public wants to know about their schools.
The framework has been built around multiple measures, which include academic, social-emotional, and school culture indicators, in order to piece together a fairer and more comprehensive picture of school performance. It consists of five major categories.
Teachers & Leadership
School Culture
Resources

LunarVim

Figure 2: Lunarvim

The gap in expertise

"All of us who do creative work, we get into it because we have good taste. But there is this gap. For the first couple years you make stuff, it's just not that good. It's trying to be good, it has potential, but it's not. But your taste, the thing that got you into the game, is still killer. And your taste is why your work disappoints you. A lot of people never get past this phase, they quit. Most people I know who do interesting, creative work went through years of this. We know our work doesn't have this special thing that we want it to have. We all go through this. And if you are just starting out or you are still in this phase, you gotta know its normal and the most important thing you can do is do a lot of work." -Ira Glass

Section 3

Learning how to debug less

With more experience, I debug less

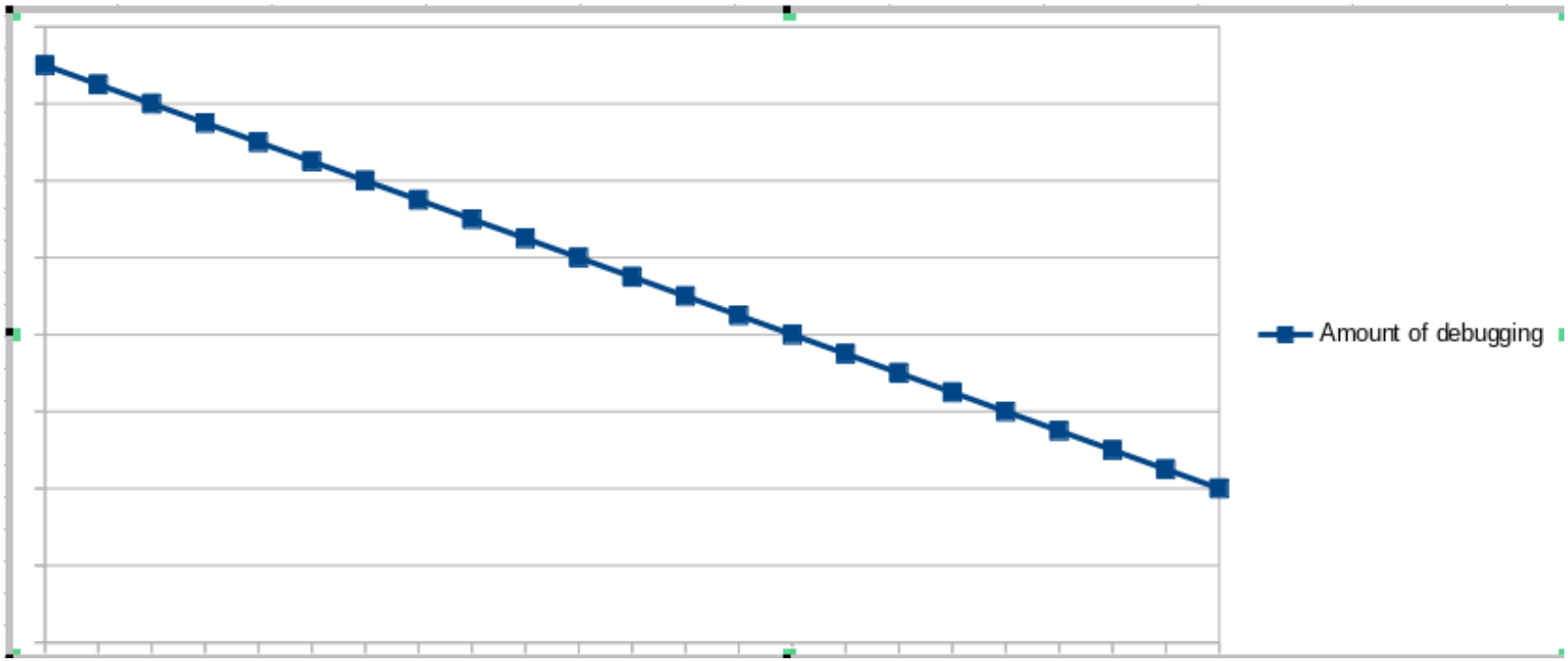


Figure 3: Amount of time spent debugging as my experience grows

The growth isn't always linear

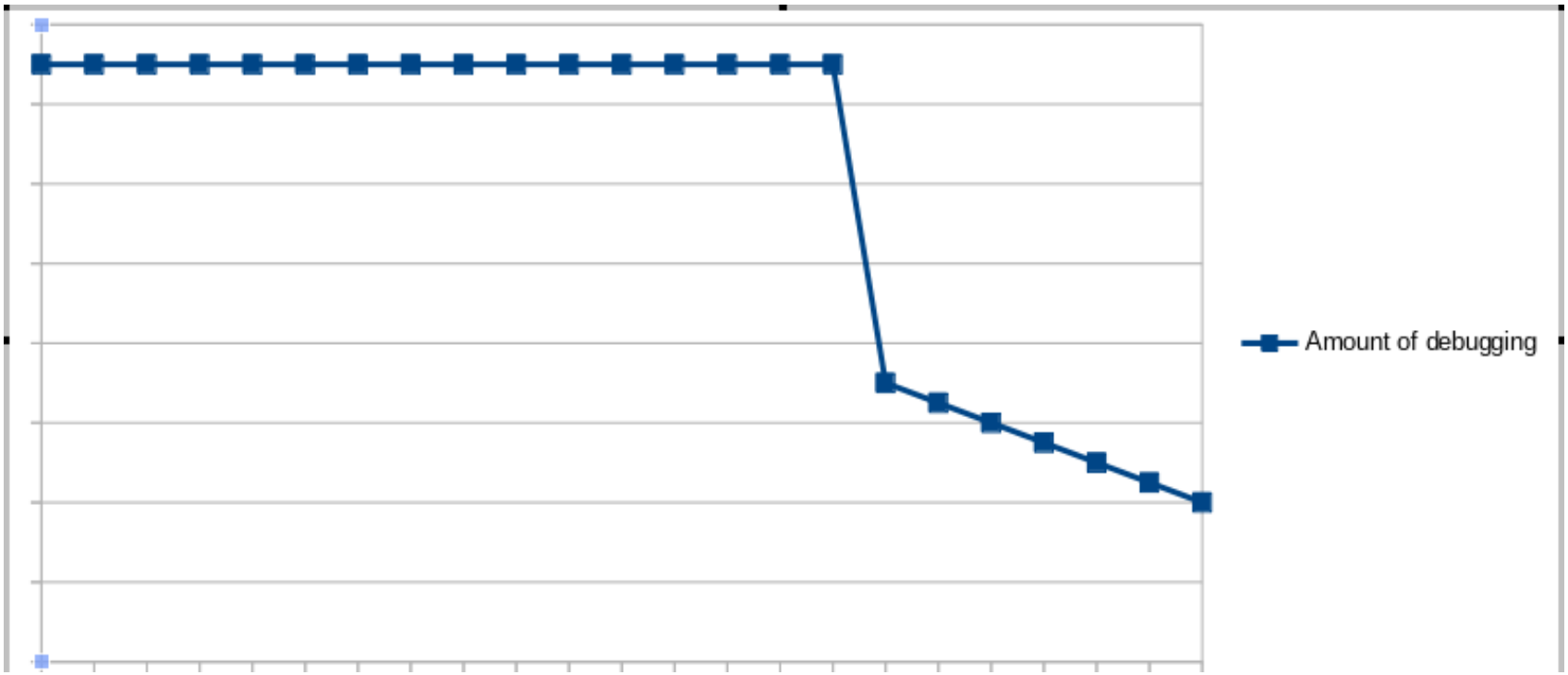


Figure 4: Amount of time spent debugging as my experience grows

What did I learn?

- Process
- Craftsmanship
- Soft skills

Amount of time spent debugging

The mythical man month, 1975

The author finds that, no matter what the intention or plan was, half of total project time is spend testing and debugging. So, rather than 'falling behind schedule', they build it into their plans.

Fallacies of Software Engineering, 2003

"The data on the percentage of time spent in error removal has varied over the years, but the usual figures are 20-20-20-40. That is, 20 percent for requirements, 20 percent for design, 20 percent for coding (intuition suggests to most programmers that here is where the time is spent, but intuition is very wrong), and 40 percent for error removal." –Robert L Glass

There are specific, actionable steps to shorten the gap

- Make small changes
- Write code that's easy to reason about
- Keep a mental model of the problem
- Make abstract concepts concrete
- When you discover a gap in understanding, reach for solutions that help complete your mental model

Section 4

Make small changes

Practice Agile development

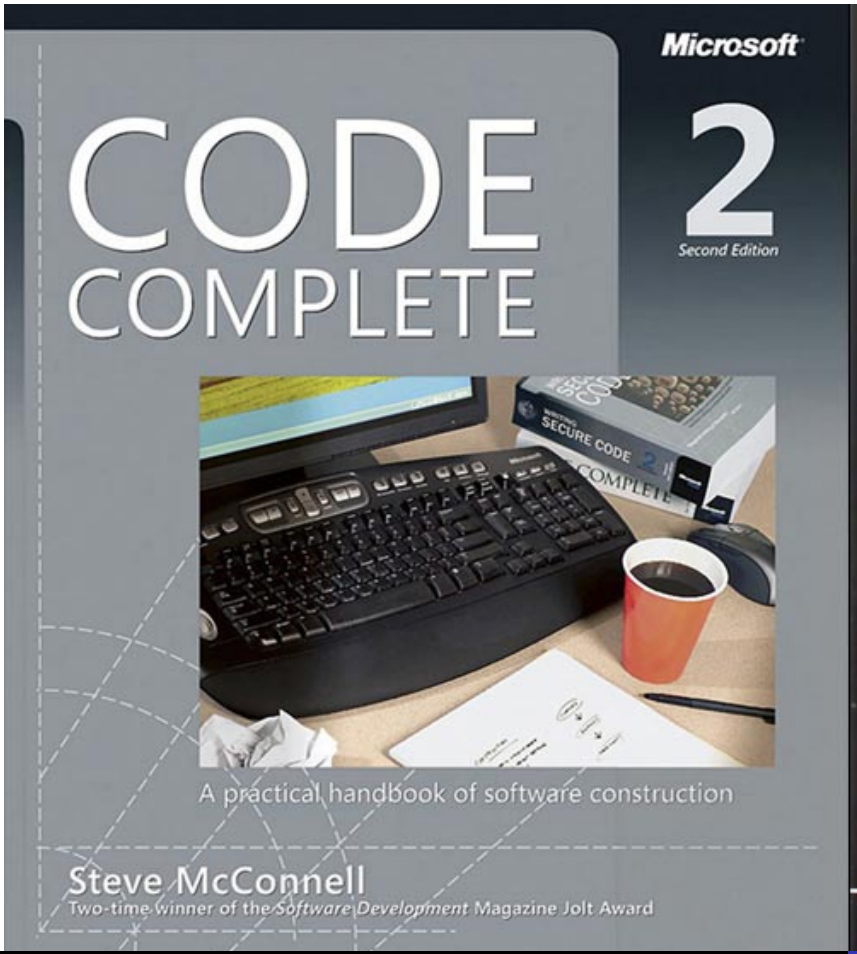
- Working software is the primary measure of progress.
- Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

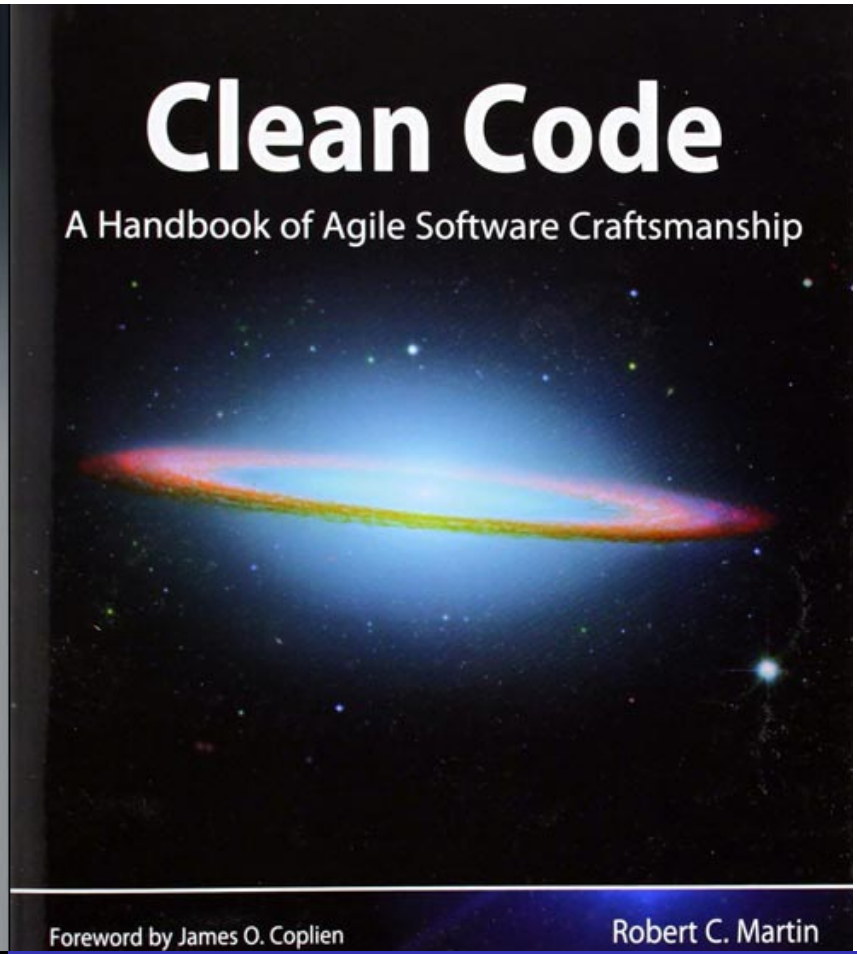
Section 5

Write code that is easy to read about

Read the classics

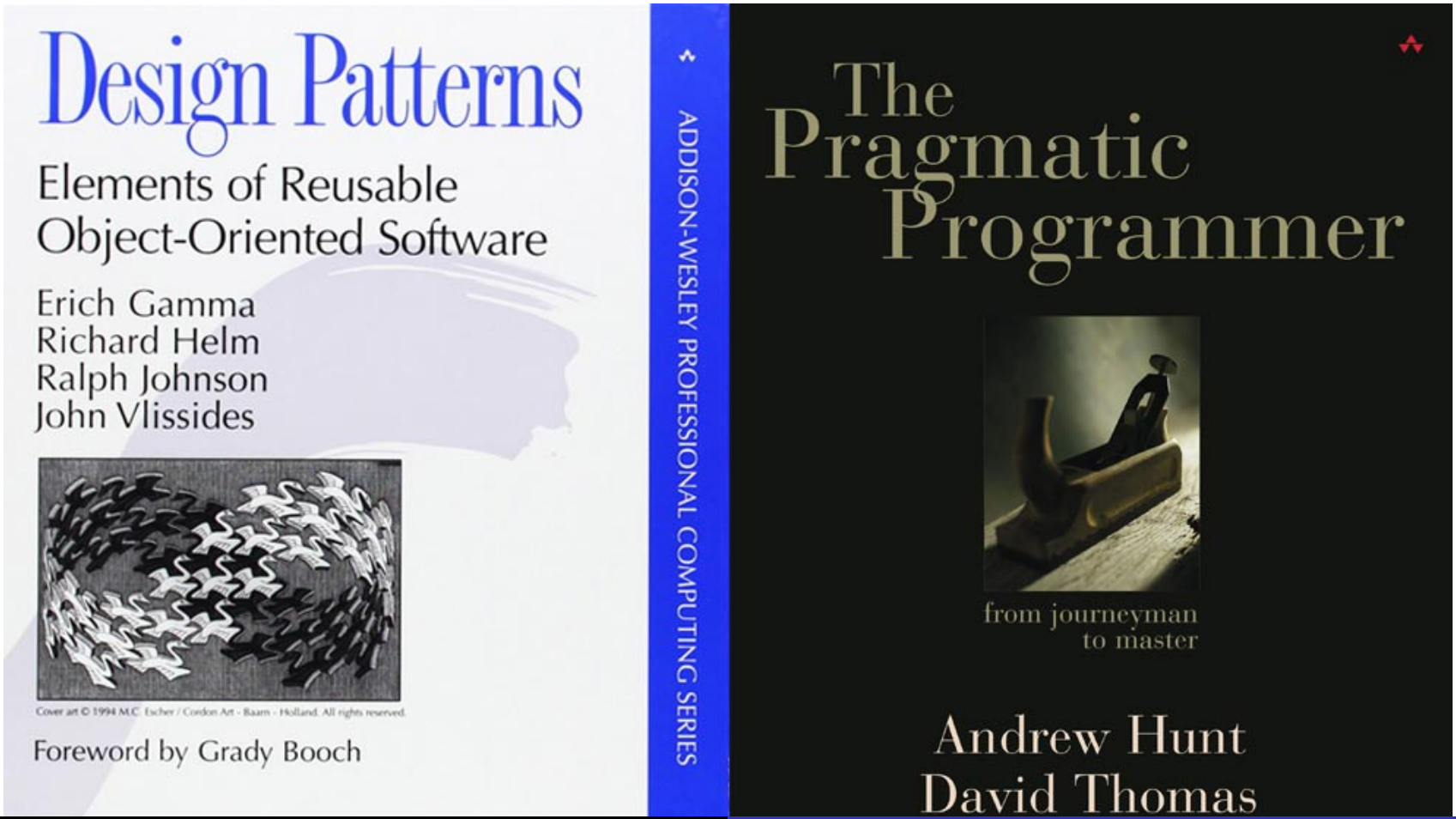
Code Complete, 2004 Clean Code, 2008



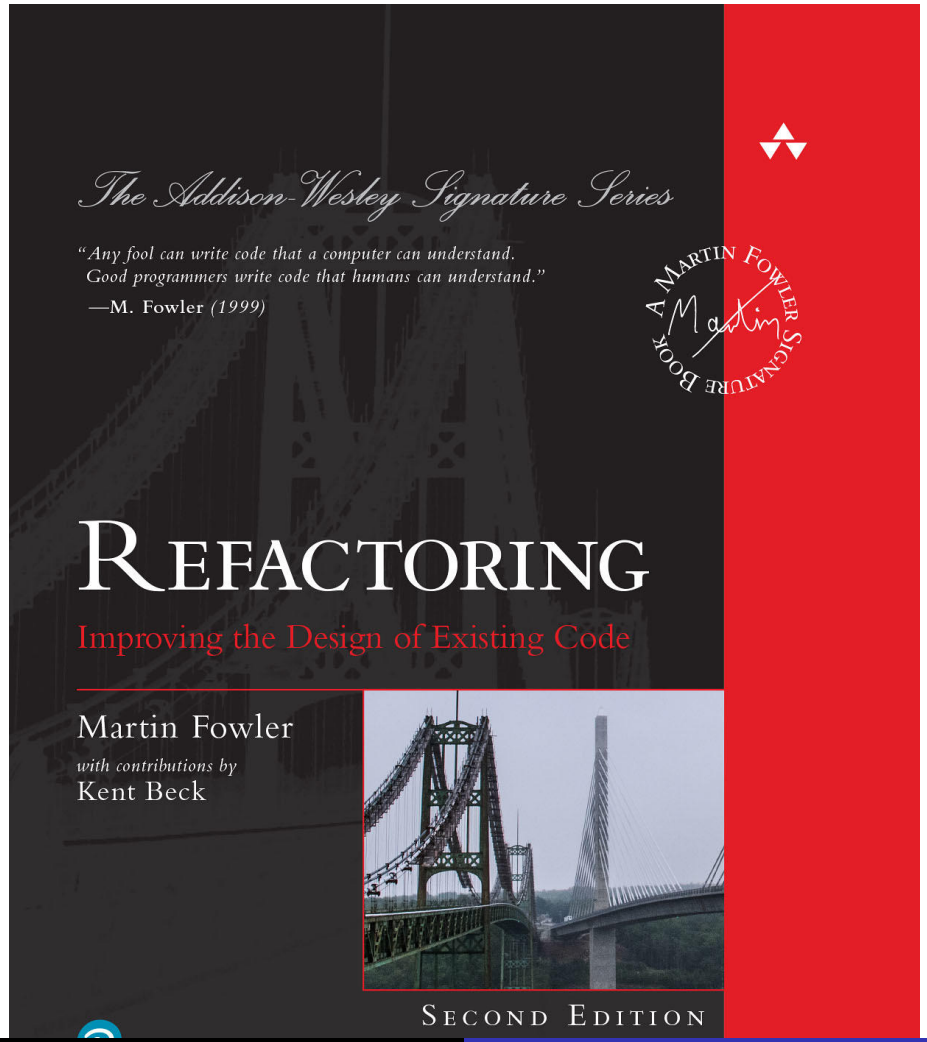


More classics

The Pragmatic Programmer, 1999 Design Patterns, 1994



Refactoring



TDD

TDD

Pair programming

Write things down

Make abstract concepts concrete

The mythical man-month

"The programmer, like the poet, works only slightly removed from pure thought-stuff. He builds his castles in the air, from air, creating by exertion of the imagination. Few media of creation are so flexible, so easy to polish and rework, so readily capable of realizing grand conceptual structures." –Frederick Brooks

Use the REPL

xpath in javascript console

Learn your tools

Readline shortcuts in vim, irb, and terminal emulator

use –help

rails g model –help

read documentation

:cheat readline :!ri Range.to_a :DevDocs google it

Section 6

When you discover a gap in understanding, reach for solutions that help complete your mental model

Rubber duck debugging

Stop coding / go for a walk. / Go to sleep

Proofread your code

Learn the debugging tools

Frontend debugging

Web console. debug gem. pry, byebug. Debug adapter protoc
Marginalia gem, better errors

debug.rb

rdbg –open=chrome

open the gem to view the source code gem open rspec

Does a github issue exist?