

Extracted from:

The Pragmatic Programmer

your journey to mastery

20th Anniversary Edition

◆ Addison-Wesley

Boston • Columbus • New York • San Francisco • Amsterdam • Cape Town
Dubai • London • Madrid • Milan • Munich • Paris • Montreal • Toronto • Delhi • Mexico City
São Paulo • Sydney • Hong Kong • Seoul • Singapore • Taipei • Tokyo

20

th ANNIVERSARY EDITION



The Pragmatic Programmer

your journey to mastery

DAVID THOMAS

ANDREW HUNT



The Pragmatic Programmer

your journey to mastery

20th Anniversary Edition

Dave Thomas

Andy Hunt

◆◆ Addison-Wesley

Boston • Columbus • New York • San Francisco • Amsterdam • Cape Town
Dubai • London • Madrid • Milan • Munich • Paris • Montreal • Toronto • Delhi • Mexico City
São Paulo • Sydney • Hong Kong • Seoul • Singapore • Taipei • Tokyo

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and the publisher was aware of a trademark claim, the designations have been printed with initial capital letters or in all capitals. "The Pragmatic Programmer" and the linking *g* device are trademarks of The Pragmatic Programmers, LLC.

The authors and publisher have taken care in the preparation of this book, but make no expressed or implied warranty of any kind and assume no responsibility for errors or omissions. No liability is assumed for incidental or consequential damages in connection with or arising out of the use of the information or programs contained herein.

For information about buying this title in bulk quantities, or for special sales opportunities (which may include electronic versions; custom cover designs; and content particular to your business, training goals, marketing focus, or branding interests), please contact our corporate sales department at corpsales@pearsoned.com or (800) 382-3419.

For government sales inquiries, please contact governmentsales@pearsoned.com.

For questions about sales outside the U.S., please contact intlcs@pearson.com.

Visit us on the Web: informit.com/aw

Library of Congress Control Number: [to come from ITP]

Copyright © 2020 Pearson Education, Inc.

All rights reserved. This publication is protected by copyright, and permission must be obtained from the publisher prior to any prohibited reproduction, storage in a retrieval system, or transmission in any form or by any means, electronic, mechanical, photocopying, recording, or likewise. For information regarding permissions, request forms and the appropriate contacts within the Pearson Education Global Rights & Permissions Department, please visit www.pearsoned.com/permissions/.

ISBN-13: 978-0-13-595705-9

ISBN-10: 0-13-595705-2

Don't Outrun Your Headlights

It's tough to make predictions, especially about the future.

► Lawrence "Yogi" Berra, after a Danish Proverb

It's late at night, dark, pouring rain. The two-seater whips around the tight curves of the twisty little mountain roads, barely holding the corners. A hairpin comes up and the car misses it, crashing through the skimpy guardrail and soaring to a fiery crash in the valley below. State troopers arrive on the scene, and the senior officer sadly shakes their head. "Must have outrun their headlights."

Had the speeding two-seater actually been going faster than the speed of light? No, that speed limit is firmly fixed. What the officer referred to was the driver's ability to stop or steer in time in response to the headlight's illumination.

Headlights have a certain limited range, known as the *throw distance*. Past that point, the light spread is too diffuse to be effective. In addition, headlights only project in a straight line, and won't illuminate anything off-axis, such as curves, hills, or dips in the road. According to the National Highway Traffic Safety Administration, the average distance illuminated by low-beam headlights is about 160 feet. Unfortunately, stopping distance⁶ at 40mph is 189 feet, and at 70mph a whopping 464 feet. So indeed, it's actually pretty easy to outrun your headlights.

In software development, our "headlights" are similarly limited. We can't see too far ahead into the future, and the further off-axis you look, the darker it gets. So Pragmatic Programmers have a firm rule:

Tip 42 Take Small Steps. Always

Always take small, deliberate steps, checking for feedback and adjusting before proceeding. Consider that the rate of feedback is your speed limit. You never take on a step or a task that's "too big."

6. Per the NHTSA, Stopping Distance = Reaction Distance + Braking Distance, assuming an average reaction time of 1.5s and deceleration of $17.02ft/s^2$.

What do we mean exactly by feedback? Anything that independently confirms or disproves your action. For example:

- Results in a REPL provide feedback on your understanding of APIs and algorithms
- Unit tests provide feedback on your last code change
- User demo and conversation provide feedback on features and usability

What's a task that's too big? Any task that requires "fortune telling." Just as the car headlights have limited throw, we can only see into the future perhaps one or two steps, maybe a few hours or days at most. Past that, you can quickly get past *educated guess* and into *wild speculation*. You might find yourself slipping into fortune telling when you have to:

- Estimate completion dates months in the future
- Plan a design for future maintenance or extendability
- Guess user's future needs
- Guess future tech availability

But, we hear you cry, aren't we supposed to design for future maintenance? Yes, but only to a point: only as far ahead as you can see. The more you have to predict what the future will look like, the more risk you incur that you'll be wrong. Instead of wasting effort designing for a uncertain future, you can always fall back on designing your code to be replaceable. Make it easy to throw out your code and replace it with something better suited. Making code replaceable will also help with cohesion, coupling, and DRY, leading to a better design overall.

Even though you may feel confident of the future, there's always the chance of a black swan around the corner.

Black Swans

In his book, [*The Black Swan: The Impact of the Highly Improbable \[Tal10\]*](#), Nassim Nicholas Taleb posits that all significant events in history have come from high-profile, hard-to-predict, and rare events that are beyond the realm of normal expectations. These outliers, while statistically rare, have disproportionate effects. In addition, our own cognitive biases tend to blind us to changes creeping up on the edges of our work (see [*Stone Soup and Boiled Frogs*](#)).

Around the time of the first edition of *The Pragmatic Programmer*, debate raged in computer magazines and online forums over the burning question,

“Who would win the desktop GUI wars, Motif or OpenLook?”⁷ It was the wrong question. Odds are you’ve probably never heard of these technologies as neither one “won” and the browser-based world wide web quickly dominated the landscape.

Tip 43**Avoid Fortune-telling**

Much of the time, tomorrow looks a lot like today. But don’t count on it.

Related Sections Include

- [Topic 13, *Prototypes and Post-it Notes*, on page ?](#)
 - [Topic 40, *Refactoring*, on page ?](#)
 - [Topic 41, *Test To Code*, on page ?](#)
 - [Topic 12, *Tracer Bullets*, on page ?](#)
 - [Topic 49, *Coconuts Don't Cut It*, on page ?](#)
 - [Topic 47, *The Essence of Agility*, on page ?](#)
-

7. Motif and OpenLook were GUI standards for X-Windows based Unix workstations.