

The Lazy Engineer: How to Think Smart and Get More Done

Mohammed Hamed Ahmed Soliman

Published by personal-lean.org, 2025.

Publisher: Personal Lean Publications
For inquiries: info@personal-lean.org

No AI-generated text, translation or images were used in the production of this work.

All trademarks and brand names belong to their respective owners and are used for identification and educational purposes only.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, including text, photographs, illustrations, tables, and diagrams, without prior written permission from the copyright holder.

While every precaution has been taken in the preparation of this book, the publisher assumes no responsibility for errors or omissions, or for damages resulting from the use of the information contained herein.

THE LAZY ENGINEER: HOW TO THINK SMART
AND GET MORE DONE

First edition. July 26, 2025.

Copyright © 2025 Mohammed Hamed Ahmed Soliman.

All rights reserved.

Written by Mohammed Hamed Ahmed Soliman.

Table of Contents

Chapter 1: The Lazy Engineer's Mindset.....	1
Chapter 2: Systems Thinking for Smart Slackers	5
Chapter 3: Automate Before You Agonize.....	10
Chapter 4: The Power of Templates and Defaults	15
Chapter 5: Energy Management > Time Management	21
Chapter 6: Lazy Problem Solving: Do Less, Think More	26
Chapter 7: MVP Thinking—How to Build Just Enough	31
Chapter 8: Lazy Leadership and Delegation — Doing More by Doing Less.....	35
Chapter 9: Engineering Out the Annoying Stuff.....	39
Chapter 10: The Lazy Engineer's Toolkit	43
Appendix A: The Lazy Engineer's Day Template.....	47
Appendix B: Lazy Engineer's Toolkit Checklist	49
Appendix C: Lazy Project Starter Pack.....	51
Appendix D: Lazy Email Templates	54

Chapter 1: The Lazy Engineer's Mindset

True Laziness is Optimization, Not Apathy

Introduction: Redefining Laziness

In most professional cultures, *laziness* carries a negative connotation. It's associated with indifference, procrastination, or avoidance of responsibility. But in the world of high-performance engineering, a certain kind of laziness is a **virtue**.

The **lazy engineer**, in this context, isn't someone who shirks duties — they're someone who **questions every step, every effort, and every wasted movement**. They find smarter, faster, and more elegant ways to get the job done. For them, doing less isn't about apathy — it's about leverage.

Bill Gates famously said, "I will always choose a lazy person to do a difficult job because he will find an easy way to do it."

This idea is at the heart of our journey.

Strategic Laziness: From Myth to Method

Consider the tale of **Larry Wall**, the creator of the Perl programming language, who once named *laziness* as a key virtue of a good programmer — alongside impatience and hubris.

Why? Because a lazy engineer will take the time to write a script so they never have to repeat a task again.

Strategic laziness is the **constant drive to reduce complexity, repetition, and waste**. It's about investing up front in thoughtfulness to avoid long-term toil.

Key Principle 1: Laziness as a Design Principle

The core question the lazy engineer asks is:

Is there a better way to do this—or better yet, a way to never do it again?

In engineering design, this shows up in principles like:

- **Fail-safes** that eliminate user error
- **Auto-calibration** in sensors to avoid frequent recalibration
- **Smart defaults** in software interfaces to reduce decision-making

Lazy engineers create systems that are *bulletproof*, *self-maintaining*, and *easy to use*. This is why automation, modularity, and simplification are more than productivity tools — they are design philosophies.

Key Principle 2: Energy vs. Time Trade-Offs

Time is measurable. Energy is not — but it is just as important.

Lazy engineers instinctively understand that 1 hour of deep effort is worth more than 3 hours of distracted work. Therefore, they prioritize:

- **Energy-efficient workflows** (avoiding context switching)
- **Tools that reduce friction** (like keyboard shortcuts or templates)
- **Mental effort savings**, even at the cost of minor time investment

Case Study:

A manufacturing engineer redesigned a testing jig so it could hold three components at once instead of one — reducing a 9-minute test cycle to 3 minutes. Time was saved, yes, but more importantly, energy and mental fatigue were reduced for the operator.

Key Principle 3: The 80/20 Rule in Engineering

Also known as the **Pareto Principle**, the 80/20 rule observes that:

80% of results often come from 20% of the effort.

Lazy engineers live by this rule. They don't obsess over perfection — they focus on:

- What gives the **highest return** on effort
- Eliminating the **bottom 80% of tasks** that produce marginal impact