

# Fixing Tools Not Problems

## Our Long Journey The Tool Treadmill Part 1

Part 2 of 7: Steps 1-3

# The Software Treadmill

## Solving One Problem Creating Another

How building software changed a long road kind of a **treadmill**.

Let's trace the steps and the pattern of added complexity.

# Step 1 Starting Out

## Raw Control e.g C

We had **full control** but wow hard work!

Had to write everything manage memory carefully.

Easy to make mistakes **slow progress.**

# Step 1 Summary

## Baseline Complexity

Problem Solved Nothing (Baseline).

New Problem **Manual Effort/Risk.**

# Step 2 Libraries Frameworks Arrive

## Reusable Code Components Helped

These helped!

Got **pre-built code** pieces.

Didn't have to rewrite basic stuff  
constantly.

# Step 2 The Downside

## Dependency Hell Framework Learning Curve

But then we had "**DLL Hell**" things  
breaking from library conflicts.

Had to learn **huge complicated**  
frameworks.

# Step 2 Summary

## Boilerplate vs Dependencies

Problem Solved Didn't have to write basic stuff.

New Problem Managing  
**dependencies and complex frameworks.**

# Step 3 Easier Languages

VB C Java Python etc

Made coding faster with **automatic memory management** (GC).

Great!

Simplified many core coding tasks.

# Step 3 The Ecosystem Explosion

## Constant Change Information Overload

But worlds around these languages got **huge**.

So much to learn **constantly**  
**changing**.

# Step 3 Summary

## Less Low Level Work More Learning

Problem Solved Less low-level coding work.

New Problem Keeping up with **giant ecosystems.**

# Insight

## Early Layers of Complexity

These first steps brought progress but also layers of overhead.

We started spending time managing the tools themselves.

# Next

## Next The Treadmill Speeds Up

Next we enter the era of Big Data

Containers K8s Microservices.

The complexity pattern continues  
and intensifies.

# Series Index

Part 1: Remember How We Got Here

**Access Part 1 PDF**

Part 2: The Tool Treadmill Steps 1-3 (Current)

**Access Part 2 PDF**

Part 3: The Tool Treadmill Steps 4-7 & Pattern

**Access Part 3 PDF**

Part 4: How AI Can Be Different

**Access Part 4 PDF**

Part 5: AI Helping Us Now Requirements Design Coding

**Access Part 5 PDF**

Part 6: AI Helping Us Now Testing Ops Roles

**Access Part 6 PDF**

Part 7: The Big Win Getting Off the Treadmill

**Access Part 7 PDF**

**Read the Full Article:** Tired of Fixing Tools Instead of Solving Problems?...

All resources mentioned are available at **<https://agilp.org/pdf/>**

[Read the Full Article on LinkedIn](#)

# Connect & Engage

LinkedIn: <https://www.linkedin.com/in/amitabhrjha/>



X (Twitter): <https://x.com/amitabhrjha>



Web: [www.agilp.org](http://www.agilp.org)



# Disclaimer & Acknowledgments

The opinions expressed are my own & don't necessarily represent my employer's views. My perspective is constantly evolving, shaped by invaluable interactions with friends, colleagues, mentors, insightful authors, and industry influencers - thank you all! Much of this content, including these carousels, is co-created with AI co-pilots like ChatGPT, Gemini, and Grok. My intent is to synthesize knowledge and share it back with the community.