Feels Like Progress

Fake Work, Fake Learning, and What Will Last When AI Can Blow It All Up

YUZHENG SUN

FEELS LIKE PROGRESS

ALSO BY YUZHENG SUN

Growth Data Analytics Playbook
(with Mengying Li and Joe Kumar)

Real Stuff (真本事), in Chinese

Feels Like Progress

Fake Work, Fake Learning, and
What Will Last When AI Can Blow It All Up

YUZHENG SUN

SUPERLINEAR PRESS

Copyright © 2026 by Yuzheng Sun
All rights reserved.

No part of this book may be reproduced, stored, or transmitted in any form without written permission from the author, except for brief quotations in reviews and as otherwise permitted by copyright law.

This book draws on the author's recollections, his earlier talks and posts, and the sources listed in the Notes. Most colleagues and other private individuals in it are not named, and some identifying details have been left out.

Photographs are from the author's collection.

ISBN 979-8-9822820-0-2 (paperback)

ISBN 979-8-9822820-1-9 (ebook)

Published by Superlinear Press, Seattle
lizheng.ai

First edition · Version 11

CONTENTS

Prologue: The Prize 1

PART ONE *What Fake Work Is*

1 The Thirty-Six-Page Report 9
2 A Metric That Measured Itself 21
3 Set Up for Failure 31
4 The Ten Percent 40

PART TWO *When AI Blows It Up*

5 Boom, Then Bust 51
6 The Default Path 62

PART THREE *Where It Comes From: Fake Learning*

7 Standard Parts 75
8 Exam Rules 88
9 The Other Side of the Table 99

PART FOUR *What Will Last*

10 Real Work Can Fail 113
11 Choose the Question 125
12 What Can't Be Written Down 132
13 Make What Lasts 143

Epilogue: A Free Afternoon 150
Afterword: How This Book Was Made 156

Notes 160
About the Author 174

PROLOGUE

The Prize

Early in my time at Facebook, I wrote a thirty-six-page analysis, structured like an economics paper: introduction, data, methodology, analysis, conclusion. Colleagues I got along with left polite likes. Almost all the comments stopped after page two.

One junior engineer read all of it. The next day, the engineer came to see me with a question: "Now that you know all of this, can you tell me what I should do?"

I couldn't answer.

That question is where this book begins. It was the first time I glimpsed something about myself that took me years to put into words. For most of my life, I had been very good at one particular game. Someone sets the question. You give the answer they're looking for. They give you a score. A good score earns you a harder question, and eventually a better school, a better company, a bigger title.

I played that game in a middle school in Yantai, a port city in northern China, and in a high school in Shanghai. I played it at a liberal arts college in America and in the economics PhD program at Cornell. Then I played it at Amazon, at Facebook, and at Tencent. By 2022, I was a vice director at Tencent, leading a data and AI team of about thirty people, collecting top performance ratings, and being prepared for a general manager role.

I don't usually lead with my résumé. If an argument needs the author's credentials to hold it up, the argument isn't finished. But in this book, the résumé is part of the evidence. It's a record of how good I was at the game, and so a measure of how long the game could hide from me what I was actually doing.

This book is about what I found when I finally took a good look at the prize.

Three moments

The thirty-six pages were the first moment. They felt like progress. I had worked hard, the methods were sound, and the document looked like serious work. It simply didn't help anyone decide anything.

The second was a conversation. In 2024, I sat down for my channel with Mike Develin, a data scientist who had led data teams at Instagram and Shopify. Mike had watched many smart technical people stall in their careers, and he traced it to beliefs they had carried out of school: that every real problem has a standard answer, that the textbook is enough, that harder things are more valuable. I recognized almost every one of them in myself. Much of what I believed about what was right at work, and what was valuable, had been formed in school, and it didn't match how the world works. Paul Graham's definition of wealth is the simplest I know: what people want. Not what a grader rewards. Only once I could see what was real could I see what was fake.

The third was a sign by the road. In the fall of 2022, outside Seattle, I passed a McDonald's with a sign out front: NOW HIRING $17.75. I asked myself, half as a joke, whether I would rather have that job or my boss's. Then I thought about it seriously, and the answer unsettled me. The promotion I was being prepared for was real. I no longer wanted the life that came with it.

Three different problems: work that changed nothing, a sense of value that school had trained into me, and a ladder leading somewhere I didn't want to go. What they share is how ordinary they are. Each can go on, in many lives, for years. And each is kept going, in part, by a sense that something is moving: the report gets longer, the grades keep coming, the title gets bigger.

I've come to believe that most of us spend a large part of our working lives inside that sense of motion. We do work that produces the evidence of work, and we mistake it for results. We learn in ways that produce the evidence of learning, and we mistake it for understanding. I'll call these **fake work** and **fake learning**, and they are connected. Long before our first job, school trained our sense of what counts as good work, and we carry it into the office. That is a large part of why fake work is so hard to see from the inside: it looks exactly like what we were taught good work looks like. Both fool the same thing, our feeling of progress, and that is why they last so long. This book is about what happens to them now, and about **what will last when AI can blow it all up**.

Why now

For a long time, the evidence of work was worth something, because it was expensive to make. It at least proved that someone had spent the time and the effort. Managers learned to read it. Schools learned to grade it.

Artificial intelligence breaks that arrangement. AI can now produce something that looks like those thirty-six pages in a minute or two. It can write the code, the report, the strategy deck, the polished answer to the exam question. It makes the feeling of progress nearly free. When work-shaped output costs almost nothing, the shape of the work stops proving much. A

polished document no longer shows that someone thought. A working prototype no longer shows that someone learned.

The subtitle of this book has a double meaning, and I intend both. To blow something up can mean to inflate it. It can also mean to destroy it. My prediction is that AI will do both to fake work. First it will inflate it: more reports, more decks, more prototypes, more evidence of effort than anyone can read. Then, over a decade or two, and unevenly, because institutions change slowly, it will destroy much of what that evidence was worth.

What keeps its value is harder to produce, and it is where this book is going. Real work is work that could fail, at something that matters to someone who depends on it. It starts with a question worth asking, because the question decides more than the effort. It is made with craft: the feel that tells an engineer, a product manager, or a writer that something is wrong before they can say why. That feel can't be written down, so no one can download it, and no machine can acquire it on your behalf. It grows slowly, through work that answers back, and it belongs to the person who grew it. And at its best, real work compounds. What it leaves behind keeps paying off long after the day it was done.

Learning and doing call for different standards. When you need to get something done, use every honest help you can find, including AI. When you're trying to learn, keep the part that matters in your own hands, because the feel grows only in the person who does the work.

What this book is

This is a book of ideas. Most of the evidence for them comes from what happened to me, because that is the evidence I can vouch for. The stories are there to show each idea at work: what it looked like from the inside, before I had a name for it.

Part One is about fake work: what it is, and why it survives so well, from a single report to an ordinary working day. Part Two is about AI and work: how AI will first inflate fake work and then empty it of value, and what that means for a career. Part Three goes back to school, where I learned the habits that make fake work feel natural, looks at what AI is now doing to learning, and follows those habits into work. Part Four is about what will last: work that can fail, the question worth asking, the craft nobody can hand you, and work that compounds.

I'm not writing as someone who has escaped the game. I did a great deal of fake work, some of it very well, and I was rewarded for some of it. I still catch myself doing it. What I can offer is an account from inside: from someone who got good at the game, kept collecting its rewards, and only slowly noticed that the scoreboard and the purpose had come apart.

Each chapter ends with a short takeaway: the idea I would most want you to keep from it.

The way this book was made puts some of its ideas into practice. If you're curious, I describe it at the end, in the Afterword. I'd suggest reading that last. By then you'll have your own judgment of the book, and what I say there will mean more.

The first chapter goes back to that document, and to the question I couldn't answer.

PART ONE

What Fake Work Is

CHAPTER ONE

The Thirty-Six-Page Report

The most dangerous fake work isn't lazy. It's diligent.

I had joined Facebook at the end of September 2019 as a data scientist on Marketplace, the part of the Facebook app where people buy and sell things with their neighbors: couches, bikes, strollers, the occasional used car. I arrived with what I thought was serious preparation. Six years in Cornell's economics PhD program. Two years as an economist at Amazon, where my job was to estimate cause and effect in situations where you couldn't run a clean experiment.

There was one embarrassing gap, and I knew about it. I couldn't write SQL, the language nearly every data scientist uses to pull data. At Amazon, someone else had usually written my queries, and I had passed Facebook's interviews in Python. So I spent my first weekends with a friend who taught me, and in two or three weeks I could hold my own. The gap I didn't know about would take me a year.

Then I went deep. I worked closely with our data engineer until I understood every database behind Marketplace and the new commerce business the company was building inside it. We built dashboards together, including the ones a group of directors reviewed every week. The first few times I presented those num-

bers, I was nervous. After a while I was comfortable, and I started to feel I understood the business.

My new-hire name badge at Facebook, September 30, 2019.

So I did what economists do once they understand their data. I wrote the paper.

The paper

An economics paper is built to survive attack. It often runs fifty or seventy pages. It opens with an introduction, then describes the data, then the methodology, then the analysis, then the conclusion. Along the way it anticipates every objection a skeptical referee might raise: robustness checks, alternative explanations, the same result tested again in different subsamples. After six years, that structure was in my bones. It was what "rigorous" meant.

My analysis covered the whole business: the channels that brought buyers in, each step of the funnel from browsing to buying, and comparisons in every direction I could think of, across

product categories, across the ways we reached users, between new users and returning ones. I looked at the business, as I put it later, from the left and from the right, from the front and from the back. The numbers held up to scrutiny. The methods were rigorous without being needlessly complicated.

I can't remember my research questions. I'm sure I posed a few at the start, the way a paper should. That tells you something. They were questions the data could answer. They weren't questions anyone needed answered.

Thirty-six pages. I posted it internally.

The response was polite. Colleagues I got along with left a like. A few people commented, and their comments clustered on the first two pages. On page three, the comments stopped. Pages four through thirty-six were, as far as I could tell, read by no one.

Except one person.

The question

The reader was a fairly junior engineer, the kind of person who loves to learn. This engineer stayed late, read all thirty-six pages, and left comments all the way through, in exactly the right places, which is how I knew every page had really been read. The next day, a meeting appeared on my calendar.

The engineer came in humble, walked through the document with me, and checked every point: this is what the data means, right? This is what your analysis is saying? I was delighted. Someone had finally read the thing, and I got to explain it.

Then came the question.

"Now that you know all of this," the engineer said, "can you tell me what I should do?"

I didn't know what to say.

It was a long time ago, and I don't remember the exact words I used to fill the silence. I do remember my first reaction, because

I'm not proud of it. It was defensive. Why was an engineer asking a data scientist what to work on? Engineers have managers for that. Assigning the work was the manager's job, not mine.

It took me a while to see that this was a dodge. Suppose the engineering manager had come to me instead and asked what the team should do. My answer would have been the same. I didn't know, and my long, careful, comprehensive analysis couldn't say either.

It could offer a great many interesting things: how conversion varied from one category to another, how one channel compared with another, how returning users differed from new ones. Each comparison was accurate. Together, they were a long list of possible explanations for why the business looked the way it did, with no way to choose among them and no suggestion of what to change.

Half an hour

The other half of the lesson took half an hour.

My manager's manager, a data science director, gave me an analysis to do. I estimated it would take two weeks, and I thought that was quick. A couple of days later, the director checked in and heard that I needed two weeks. The director had expected a day. Rather than argue the point, the director showed me. We sat down together, and while I watched, the director opened a blank Google Doc and a blank Google Sheet, wrote some SQL, and did the analysis in the spreadsheet. In about half an hour it was mostly done. Then it became a short note, posted for everyone to see.

The analysis was, frankly, simple reporting. It compared how well our subsidies were working this year with how they had worked the year before, and it found they were working better.

It caught the attention of the vice president who ran commerce on Marketplace, because the director knew it answered a question that vice president cared about a great deal.

Years later, when I finally told this story on my channel, I described my two-week version this way: it was a fantasy in my head. I had imagined I was doing something impressive, and I was performing elaborate moves for an audience of one. Nobody cared.

The lesson was not "work faster," and it was not "do less." The half-hour analysis wasn't better because it took less time. It was better because the director started from what people needed to know and answered that. My two-week plan started from what I found interesting, and from what I had been trained to believe a serious analysis looks like. Given where each one started, it wasn't that my two-week version might turn out worse. It was bound to.

What it felt like from the inside

Those thirty-six pages felt like progress, and that feeling is what this book is about. Every page was a small piece of evidence that I was working hard and working well. The hours were real. The SQL was correct. The methods were sound. The polite likes seemed to confirm it. If you had asked me that week whether I was doing good work, I'm fairly sure I would have said yes and shown you all thirty-six pages.

None of it was laziness. When people hear "fake work," they picture someone hiding from the boss: scrolling on a phone, stretching a task to fill the afternoon. That kind at least knows what it is. The kind I was doing is often the most careful, most rigorous, most work-like work in the building. It produces exactly the signals we have been taught to read as quality: effort, length, polish, completeness.

Paul Graham saw this years ago. Graham called fake work, not fun, "the most dangerous way to lose time." The reasoning was simple. When you spend a day watching TV, an alarm goes off, and by evening you know you've wasted it. When you spend a day dealing with email, no alarm goes off, because it looks like work. Fun doesn't pretend to be progress. Fake work does. That's why it fools not only your boss, but you.

Fake work, as I define it, is visible activity standing in for value that is hard to measure.

The definition says nothing about effort and nothing about form. You can't identify fake work by looking at it. A report that changes the next decision is real work. A nearly identical report, written to show that the team is thorough, is fake work. The same goes for meetings, dashboards, decks, and code. A feature nobody needs, shipped on time and free of bugs, is just a more expensive kind of fake work.

What separates real from fake is what the work is for. Follow it downstream: whose decision or action is it supposed to change, and could anyone tell whether it did? An investigation that concludes nothing should change is still real work, because a decision rests on it. A report that no decision will ever rest on is not.

Fake work describes an activity, not a person or a job. A useful job can be full of it, and someone stuck in a role that shouldn't exist can still do something useful. I care about the activity, because the structure that produces it deserves the criticism, not the person caught inside it.

Why nobody told me

Why didn't anyone tell me the report was useless?

Because almost nobody ever does. This is one of the most important facts about fake work. **It doesn't get punished. It gets skimmed.**

Think about the feedback I received. A handful of likes from people who were friendly with me. Comments on the first two pages. Silence after that. None of it said, "This doesn't help anyone." All of it, if I wanted to read it that way, said, "Nice work." And I wanted to read it that way.

This isn't a failing of any particular person. People are busy, and they're decent. Telling a new colleague that a thirty-six-page document doesn't help anyone is awkward. It costs you something and gains you nothing. Skimming it and leaving a like costs a few seconds. Multiply that by everyone in a company, and you get a system in which fake work receives a steady trickle of mild approval, indefinitely.

If this sounds familiar, it's probably because of all the AI slop: long, polished documents that nobody reads past page two. That isn't a new problem. It's this one, at a far larger scale. I'll come back to it at the end of this chapter, and again in Part Two.

The most useful feedback I got came from the one person who took the document seriously enough to read every page. Fake work survives on skimming. The closer anyone looks, the more exposed it becomes. And the closest kind of attention isn't checking whether the numbers are right. It's asking what to do about them.

From A to B

It took me about a year to rebuild how I worked, one small correction at a time, mostly by watching how the strongest data scientists at Facebook did it. The paper format was the first thing to go. I started paying close attention to data visualization. One chart tells one story, not several. You don't put a chart in front of people and leave them to work out what to read; you add an arrow and a note that tell them how to read it. A presentation

needs a purpose before it needs slides. And you start with a story or a metaphor, something people will remember.

The weekly leadership review was a demanding teacher. The directors looked at the numbers every week, and the product managers we worked with moved fast. One especially driven product manager, if a data scientist didn't deliver a requested number the same day, would simply write the query. No journal referee had ever put that kind of pressure on me. It meant that my value couldn't be producing analysis. Other people could produce analysis, and some of them would do it themselves if I was slow. If I kept telling people useful things about what they cared about, and what to do about it, they would pay attention, even to something I had made in half an hour.

At Facebook's Seattle office on March 4, 2020, days before it closed for the pandemic. The rocking-horse poster at top left says "Don't mistake motion for progress."

Eric Gregory, one of the most senior data scientists at Facebook, put it more bluntly than I could, in a LinkedIn post I've paraphrased many times since. A data scientist's value, as

Gregory described it, is this: everyone was going to do A; you show them they should do B; B is better than A. The improvement from doing B instead of A is your value. That's it.

I would add one thing. In a modern tech company, nearly all work comes down to one of two things: you build, or you influence a decision. Whatever the job title, and however complex the work looks, it comes down to these two. Engineers build the product. A data scientist like me mostly influenced. If my work didn't change what someone else decided or did, it didn't change anything, no matter how good it was.

That is true far beyond data science. A strategy memo, a market analysis, a slide deck for the leadership offsite: all of it is influence work, and each piece is real exactly to the extent that it changes what somebody does next. The engineer's question, "What should I do?", can be asked of almost any of it. A surprising amount has no answer.

Today I describe the shift I went through that year this way: I moved from being responsible for the report to being responsible for the result.

Later, short analyses like that director's, together with some of mine, helped turn the whole organization in a new direction. I'll tell that story in Chapter 11. What matters here is their shape. They were short. They told people to stop doing A and start doing B. And they could have been wrong.

Thirty-six pages for free

I moved to Tencent in 2021, and there I met the thirty-six-page report again, many times over.

Tencent has an internal knowledge platform where employees post articles, and its data science corner was full of them. Many of the data analysis articles I read there followed the same pattern: comprehensive, methodical, impressive, and of almost no

practical use. A typical one would lay out ten ways to find a product's "aha moment," the point at which a new user turns into a loyal one. Step one, step two, how to spot the kink in the curve. You'd finish it knowing ten methods. You still wouldn't know what to do on Monday.

The idea itself is shaky. Take the most famous example, the claim that new Facebook users who add ten friends in their first week are the ones who stick around. Users who reach a number like that were usually more eager to use the product to begin with. The moment predicts who will stay; it doesn't show what makes people stay. Push everyone toward it, and much of your effort goes to people who would have stayed anyway. Data scientists love writing these articles because they seem to turn the most important question a product has, why people stay, into steps anyone can follow. The steps are tidy. The foundation is a correlation mistaken for a cause.

Then, at the end of 2022, ChatGPT arrived. When I saw what it could do, my first feeling was relief. Thank goodness I had already gotten past that stage. ChatGPT could produce the thirty-six-page report. Not my exact report, but its essence: a long list of possible answers, carefully organized. Ask it why your conversion rate is low, and it will give you ten reasons and a dozen comparisons in less than a minute.

My second feeling was less comfortable. Most of the analysts I had watched over the years were still in that stage. And now a machine could do what they did.

Once you can tell people what they should do, and be right often enough that they listen, you are hard to replace. Until then, you are competing with a machine that writes thirty-six pages for free.

The thirty-six pages were never worth much. AI didn't change that. It only made it visible. Their cost made them feel valuable, at

least to me. That is where everything AI does to fake work begins: it strips away the cost that used to disguise it.

Being fair to my younger self

My younger self wasn't careless. He was doing exactly what he had been trained to do, and doing it well.

For twenty years, the people who evaluated me had rewarded the qualities that thirty-six pages put on display. Thoroughness. Rigor. Covering every case. Anticipating every objection. Showing your work. In a classroom, those are the qualities that earn the grade. In a PhD program, they are how you survive a referee. I had learned to treat work as an exam.

At Facebook, the questions that mattered weren't handed to me, and nobody was going to grade my answers for completeness. What mattered was whether something changed. That was a different game, and I was playing the old one with complete commitment.

Only much later did I understand why my sense of what was valuable at work had been so far off. It came from how school is designed. An exam can reward only what it can check, so it checks what an answer looks like: whether it is complete, whether it follows the method. It almost never checks whether the answer changes anything. By every standard I had, the thirty-six pages were good work. That is why I couldn't see what was wrong with them. It is also why this book can't explain fake work without explaining fake learning. The standards we use to judge our work were set in school, so seeing fake work clearly starts with seeing where those standards came from. Part Three goes back to where I learned the old game.

The rest of Part One asks why fake work survives. My thirty-six pages survived because nobody told me they were useless, and that is the easiest case to see. The next three chapters follow

fake work to places where it survives for deeper reasons: a team whose own number says it is winning, a goal that the people above can read any way they need, and a working day that simply feels full.

Takeaway

The most dangerous fake work is diligent: careful, thorough, and aimed at nobody's decision. It survives because nobody tells you it's useless. It doesn't get punished. It gets skimmed. So don't wait to be told. Ask of anything you make the engineer's question: because this exists, who should do what differently? If you can't name the person and the action, you have your answer.

CHAPTER TWO

A Metric That Measured Itself

A dozen scientists lost to the median of six numbers.

Not long after I left Amazon, I heard how it happened. A team at Amazon Fresh, the grocery business, had more than a dozen applied scientists building deep learning models to forecast one of its most important numbers. At a monthly business review, an economist from a neighboring team listened to their forecast and pointed out that the data had only about three hundred points in it. Then the economist offered an alternative: take the median of the last six data points, and use that as the forecast.

The scientists were insulted. The senior leader running the review decided to settle it: from then on, both forecasts would be compared at every review. The median won six times in a row.

I don't believe the scientists were surprised. Three hundred data points are far too few for deep learning, and among more than a dozen scientists, someone knew. So why build it? I think because the model was doing a different job. Sophisticated work is good for the scientists, who are paid well to do it and can write papers about it. It is good for a director, who now runs a science team, and for the vice president above, who can argue for promoting that director and compete with other vice presidents for headcount and scope. Each layer had a reason to want the model. None of those reasons was a better forecast.

That story put words to a question I had been asking since my first year at Amazon: what is all this serious work actually for? The people around me were capable, and they worked hard. But much of what they did seemed to exist to justify something: a goal, a plan, a team, a number. My own team was a quieter case of the same thing: a team measuring its progress against a number it had defined itself.

I had joined Amazon in 2017, after finishing my PhD, as an economist on the team that fought counterfeit products. People are often surprised that Amazon employs economists at all. At the annual job market for new economics PhDs, a university might hire two or three. Amazon hired dozens.

The reason is that Amazon sells real things for real money. Facebook or Google can show two groups of users two versions of a free product and see which works better. When a change affects what customers pay or how sellers are treated, Amazon often can't, because a randomized experiment can raise questions of fairness, or of law. So it needs people who can estimate cause and effect without one. That was my job.

I was building models, and the models were interesting. You could make them quite sophisticated and present them to people. But I came to feel that the model didn't mean anything, and then that most of what our team did didn't mean anything either.

Two numbers

Our headline number counted only what our own models could see. Every team at Amazon had a headline metric, the number others judged it by. Ours counted bad orders: counterfeits and other products with serious problems that had reached customers. The goal was to bring it down. But the only bad orders it counted were the ones our machine-learning models detected, on the order of a hundred thousand.

So our models decided what the problem was, and the same models decided whether we were solving it. Whether a model was any good, and whether the team's work was working, were both judged by a number the team produced itself. You could see how strange that was whenever a model improved. A better model caught bad orders it used to miss, and the number went up, as if things had gotten worse for customers. They hadn't. Those bad orders had been there all along; we just hadn't counted them.

There was a second number, and it came from customers. When someone returns an item on Amazon, they choose a reason for the return, and one of the reasons says the item was defective. Nobody on our team decided what went into that number. Customers did, one return at a time, and those returns numbered in the tens of millions. No change to our models could move it; only a change in what customers actually received could. That number was the problem itself, as customers lived it.

The gap in size showed how little of the problem our models could see: two orders of magnitude. But what mattered more was that the second number was out of our hands. If our team existed to protect customers from bad products, I thought, the second number was the one we should be trying to bring down.

This was not my problem to solve. I was a junior economist. Choosing a team's headline goal was the business of product managers and senior leaders, several levels above me.

But I couldn't let it go. I brought it up with my manager again and again, almost every day, until my manager gave in. There was no stopping me anyway, I was told, since I was clearly going to keep bringing it up. So I got twenty percent of my time to pursue it. Officially, I kept to twenty percent. Most of my attention, honestly, went there.

I went looking for anyone who could tell me something. I talked to people who ran the warehouses. I talked to the supply

chain optimization team. I talked to customer service. I talked to the senior people on my own team, to our product managers, and to people on neighboring teams. Each conversation refined the argument, and each time I went back for more data to support it.

This made me unpopular in a specific way. Some of our product managers were annoyed, and they had a point by the logic of the organization. Deciding what the team should measure was a product manager's job. Who was this economist to go around the building arguing about it? And why, after the team had already agreed on its goal, did this economist keep pushing? It was disruptive. It got in the way of people's normal work.

In the end, the argument won. The metric in the tens of millions became our team's S-team goal, replacing the one in the hundreds of thousands. The S-team is Amazon's senior leadership team, and an S-team goal is about as high as a goal can go inside the company. Our team's scope grew a great deal.

My own standing did not. I had become, as I put it later, the destabilizing factor. People recognized the effort, but they didn't like the behavior. I don't hold that against anyone. I don't think my manager or my team did anything wrong. In a large company, what I did really is disruptive. It makes the boss's job harder and everyone else's job harder too. I accept that.

What stays with me is how hard it was, when the answer had been sitting there all along. Customers had already told us which orders were defective. Nobody had to invent anything. But the organization was set up to play a different game: pick a number, push it, and measure yourselves against its progress. The progress our team had measured against its own number felt real to everyone. It could move without anything changing for customers, and in that sense it was fake.

Everything felt hollow

Most of my work at Amazon never landed anywhere. The metric story has a result at the end. Day to day, my two years there felt different, and that is what eventually pushed me out. The metric wasn't the only way our team ended up judging its own work. The other way was argument.

The word I kept using for my work back then was the Chinese word *xu*. It means hollow or insubstantial, and it also means abstract: something that stays in ideas and never becomes concrete. It's what you call a punch that never lands. That is what most of my work felt like. We discussed things, and analyzed them, and discussed them again, in a kind of vacuum. Very little of it ever landed in Amazon's actual product.

Part of the reason was structural. Scientists and engineers sat in separate organizations. Anything we wanted built had to be discussed, argued for, and then put on the roadmap of an engineering team next door. I had to go up my own chain until someone senior enough agreed it was worth doing, and then that person had to negotiate with an engineering director, whose team always had more to do than it could do. The discussion alone could take weeks or months. Even when everyone agreed, the engineers often didn't make our work a priority, and often couldn't. So a great deal of what we made had no channel through which it could reach the product.

Because many decisions at Amazon couldn't be settled with an experiment, they were settled by argument. Whether something was worth doing came down to persuasion and judgment rather than a credible result. The effect, as I described it at the time, was that you spent most of your time getting the project approved rather than doing the project.

Once, it seemed like my work would finally land somewhere. A neighboring team had launched a program called

Transparency, which put scannable codes on products so that their origin could be traced. My model could have helped. I talked it through with their side. The product managers liked it, and so did the engineering manager. But the director on their side was new, saw it differently, and said no. That was the end of it.

The small things told the same story. The hardware security keys we used to log in were still the old USB-A kind, though many of our laptops had moved to USB-C, so you had to buy an adapter. It made no sense. I mentioned it, and nothing happened. Fixing it meant going up the chain several rounds and back down again, and for any single person who might do that, the payoff wasn't worth the trouble. So nobody did.

Why organizations do this

Inside a company, nobody can see value directly. At the time, I mostly blamed the process: too many layers, too much case-making, too little action. That was part of it. But the case-making and the metric were two faces of that one problem, and neither was really about bureaucracy.

Economists have understood this for a long time. In 1937, Ronald Coase asked why firms exist at all. If markets are so good at coordinating activity through prices, why don't we all just trade with one another as independent contractors? Coase's answer was that using the market has costs: finding partners, negotiating, writing contracts, enforcing them. Some activities are cheaper to coordinate by direction, inside an organization, than by negotiating prices in a market.

That bargain has a hidden cost. Inside the firm, most work doesn't come with a price attached. Nobody's afternoon arrives with a number that says what it was worth. A market tells a bakery every morning, in cash, whether people want its bread. Nothing tells the anti-counterfeiting team, every morning,

whether customers are better off. The organization has to guess, and it guesses through proxies: reports, dashboards, launches, headcount, and above all metrics.

Proxies are necessary. You can't run a company of a million people without them. The trouble starts when proxies decide budgets, promotions, and status. Then people begin producing proxies.

In a well-known model from 1991, the economists Bengt Holmström and Paul Milgrom showed what happens when a job has several parts and only some of them are easy to measure. If you reward the measurable part strongly, effort tends to flow toward it and away from the parts that are harder to measure, which can include the parts that matter most. Nobody needs to cheat for this to happen. People simply put their effort where it is seen.

What I saw at Amazon was one of the clearest versions of that problem I've encountered. It wasn't just that the measurable part of the job got the attention. The team was judged by a number its own models produced. And I don't think anyone chose it cynically. A metric defined by your own model has real advantages. You have the data. It responds quickly to what you do. It's easy to explain. Most of all, it's easy to claim: when it falls, the credit seems clearly yours, even though a fall could just as well mean the model has started missing things. The customer-level number is the opposite. It is huge and noisy. It depends on sellers, on warehouses, on customer service, on a dozen other teams. If it improves, who gets the credit? If it gets worse, who gets the blame?

So teams drift, without anyone deciding it, toward metrics they control. And a metric produced by your own system has a quiet flaw: it can move because something changed in your system rather than in the world. It can become a mirror.

The team-sized version of the feeling

A self-measured metric gives a team the same reassurance that a long, careful report gives the person who wrote it. Picture how it works. The models flag bad orders, other teams act on them, and next quarter the models flag fewer. The line on the dashboard moves the right way. Reviews go well. The feeling of progress is real and widely shared, and it is renewed every quarter.

What it doesn't require is that anything changes for the people the team exists to serve. The loop closes inside the building.

A self-measured metric isn't always wrong. Sometimes the model's count really does track what customers experience, and bringing one down brings down the other. The question is whether anyone has checked. In my experience, it is surprisingly rare for a team to ask how many steps lie between its headline number and a real change in a customer's life.

Changing the metric

When a metric points at the wrong thing, changing it is much harder than hitting it, and usually worth far more.

Hitting a metric is execution. Someone has already decided what matters, and you work to move the number. That's valuable, and it can be very hard. But changing the metric is deciding what the work is for. When the metric changed from a hundred thousand to tens of millions, the same team was suddenly pointed at a far larger problem, and one defined by what customers reported rather than by what our models counted. A better model now helped the number instead of hurting it. That alone doesn't prove customers ended up receiving fewer defective orders; changing the scoreboard is a beginning, not a result. But it changes what every hour of effort is aimed at.

It is also socially expensive, which is why it so rarely happens. Changing a team's metric implies that the old one was the wrong

target. It makes people's past success look smaller. It reshuffles who owns what. It is usually not your job. Whoever pushes for it will usually be unpopular for a while.

Nobody should go looking for a perfect metric, either. Knowledge work will always need proxies. But every proxy should be checked, now and then, against what customers actually do, and every proxy should be allowed to expire. A metric that can never be retired will eventually become a machine for producing fake work.

Years later, I found useful words for the difference in Michael Gerber's *The E-Myth Revisited*, a good book held back by its title. Of the personalities Gerber sees at work in anyone who runs a business, two capture what I mean. The *technician* gets things done: hand a technician a well-defined problem, and it will be solved. That is a real virtue. The *entrepreneur* keeps asking whether the thing being done is the thing worth doing. Most organizations are built to reward technicians. They need entrepreneurs, and they find them annoying.

I wasn't trying to be an entrepreneur at Amazon. I just found it unbearable to do something stable and meaningless, and I was only happy when my attention was on something that mattered, frustrating as that often was. I've come to think this feeling can be a useful signal, not just a quirk of personality. It may be the part of you that is still asking what the work is for.

Why I left

I left Amazon after two years. The reason was simple: I wanted my work to touch a product. Facebook had a reputation for the opposite of what I'd experienced. An engineer could have an idea one day, have it running in an experiment the next, and see the results two or three weeks later. I wanted that badly enough to take a title that sounded less impressive, data scientist instead of

economist. I also took a cut in pay, because Amazon's stock grants paid out mostly in the third and fourth years, and I left early in my third.

I got what I asked for, though not the way I expected. Soon after joining Facebook, I had plenty of product impact. A lot of it was negative. And one of the first things I learned there was that a metric can do worse than measure the wrong thing. It can be used as a weapon.

Takeaway

Much of the serious work inside an organization exists to justify something: a goal, a team, a number. A team judged by a number its own work produces has defined its own success, and its progress can be fake while everyone works hard. Find the number nobody inside controls, usually the one customers already give you, and measure yourself against it. Changing the number is harder than hitting it, and worth far more.

CHAPTER THREE

Set Up for Failure

"A man always has two reasons for what he does—a good one, and the real one."

—Often attributed to J. P. Morgan

The Chinese historian Wu Si built a whole book on the same observation. In *Hidden Rules*, he argued that officials in the Ming dynasty lived by two sets of rules. The official rules were the ones everyone said out loud: filial piety, benevolence, loyalty to the throne. The hidden rules were the interests underneath, and they decided most of what actually happened. Everyone knew both sets. Only the first could be spoken.

I first saw the two reasons clearly in my first winter at Facebook. The team I supported hit both of its goals. It was given a very poor rating anyway, and within a few months most of the people who ran it were gone. The good reason was about the metrics. The real reason, as far as I could tell, had little to do with them. The metrics were the tool.

In October 2019, while I was still at new-hire training at Facebook's headquarters in Menlo Park, the organization I was joining went through a major overhaul. A senior leader had brought in a group of people from outside the company to build a

traditional e-commerce business inside Marketplace, and the people who had built Marketplace into a thriving place for neighbors to buy and sell used goods found themselves pushed to the edges. You can read that as a takeover, or as a strong mandate in a hurry, with no time to win over people who didn't believe in it. I've never been sure which reading is closer to the truth. Probably some of both. Either way, I had joined a team in the middle of it.

The Facebook Wall at headquarters in Menlo Park, October 2019, while I was still in new-hire training.

The goal

The team I supported went into the holiday season with two goals, agreed at a leadership review. The first was that total transactions would hit a target. The second was that week-over-week retention would grow: a larger share of the people who bought something one week would come back to buy the next.

I raised a warning about the second goal. Retention growing during the holidays doesn't mean the team caused it. People shop more in the holidays. Retention could go up even if the team did nothing at all. The clean way to know whether the team's work was responsible was to run an experiment, holding some users out and comparing.

Everyone in the room understood the point. But it was too late to set up that kind of experiment before the season began. My manager had joined only a month or two before me and, coming from elsewhere, hadn't known to set one up. I don't blame my manager for that. The product leader agreed, the lead product manager agreed, and so did everyone else: given the circumstances, we would track the goals as written and live with the caveat.

The team delivered. It beat the transaction target by a wide margin, without spending much money. Week-over-week retention grew too.

Then, as I remember it, the same caveat was turned against us. The result wasn't causal, the people above us said. How had the goal been set? Who had set it? The holiday season would have produced those numbers anyway.

All of this had been said, out loud, in the meeting where we set the goal. It didn't matter. The engineering manager pushed back, and that didn't matter either. The goal had been poorly set, the argument went, and so the result didn't count. And whose fault was a poorly set goal? Data science's. Which data scientist's? My manager's? I had been there a couple of months. I was not going to stand up and blame my new manager.

A good manager, I thought then and think now, would have stepped forward and taken that blame. But I could also see that my manager was new and struggling to stay afloat. Somebody had to absorb it. So I did. I said yes, the goal had been poorly set.

Cleared out

The team that had hit its goals was given a very poor rating. Over the following months, the product manager, the engineering manager, most of the senior engineers, and the engineering director all left. In effect, the team was cleared out.

I survived for a reason that had nothing to do with merit. I was too new to evaluate. New employees didn't receive a rating in their first half year, and I didn't get one. If I had, it would almost certainly have been bad, despite the overtime, the late nights, and the volume of work.

The engineering manager never blamed me. Later, the engineering manager and the product manager urged me to go to my manager and say plainly that the goal had been set badly, to put the responsibility where it belonged. I didn't. I didn't feel I could.

The good reason

The team would probably have hit the retention goal even if it had done nothing, because the holidays would have done the work. **A goal you will hit regardless of what you do is not really a goal. It's a weather report**. And hitting it produces a powerful feeling of progress, for everyone involved, with no information about whether anyone's work mattered.

Nobody on the team chose fake work. The team worked brutally hard. The engineers built real features. The transactions were real transactions. But the scoreboard couldn't tell the difference between the team's effort and the calendar.

A number that can't settle anything can be used to say anything. When the team hit its goal, the number could have been read as success. When the people above wanted a different verdict, the same number, with the same flaw that everyone had acknowledged in advance, could be read as failure. **The metric didn't decide the outcome. It supplied the good reason for an**

outcome decided some other way. The real reason was never said out loud. I can only guess at it, and my guess is the darker reading of the reorganization: the new leadership wanted its own people running the business, and the team that had built it was in the way.

That winter taught me something no exam had. A metric isn't only a measuring instrument. In the hands of people with power, it is also a way to announce what they have already decided, and to make it sound like arithmetic.

Two jobs

When the number doesn't decide the verdict, the people being judged notice, and they adjust.

Anyone who has worked for a while knows that doing valuable work and proving that the work is valuable are often two separate jobs. When the scoreboard can tell good work from luck, the second job is small, because the result does most of the proving. When the scoreboard can't, the second job decides the verdict, and the second job is producing good reasons.

That is how a structure manufactures fake work without anyone choosing it. Goals get set as numbers that can be hit without anyone being able to say why. Then the same numbers are used to reward or punish, depending on what the people above need. People below learn, quite reasonably, that the work itself isn't what's being judged. The story is. So, over time, more of their energy goes into the story: how a goal is worded before the season, and how a result is explained after it.

Big companies have their own version of Wu Si's two sets of rules. Out loud, the language is customer obsession and connecting people. Underneath, the conversations are about scope, visibility, headcount, and the next promotion. Most of us don't see the second set when we start. As we rise, we do, and we split:

some decide to play the hidden game well, and some decide they don't want to play it at all. That winter was when I first saw it, and it wasn't the last time.

This is why I'm wary of advice that treats fake work purely as a matter of personal discipline. "Just focus on what matters" is good advice, but it assumes the environment will eventually notice. Sometimes it can't, because its scoreboard can't see.

The hardest months

Inside that structure, effort stopped telling me whether I was doing well.

Those months were the hardest of my working life. A new product lead arrived who did not trust me at all. When I spoke in meetings, I could practically see the eye-rolls. Everything I did was wrong. If I did more, I made more mistakes. If I did less, that was wrong too. This product lead often worked until two or three in the morning. I usually worked until ten or eleven at night just to get my queries to finish and my reports written. Then, at midnight and at one, the questions would start arriving. Even if I told myself I'd answer in the morning, my mind would keep running. I might fall asleep at three or four. I'd wake at seven or eight, think about work, and be unable to fall back asleep.

I made mistakes I couldn't have known I was making. Once, an experiment I was supporting showed wonderful results. Then people from other Marketplace teams came over to point out that we had made our part of the screen wider and squeezed everyone else's. Our numbers were up because theirs were down. They were right, and we took it down. But the team had had no data scientist for about a year before me, and there was no one to tell me the context. I was learning the rules of a game by being penalized for breaking them.

I am, by temperament, someone who doesn't complain. That fall I complained to everyone: how fast everything moved, how high the expectations were, how everything I produced turned out to be wrong. I spent Thanksgiving in Iceland, on a trip booked before I'd even started at Facebook, and lay awake there too.

That was when I started drinking to fall asleep. Before then, I had never really understood the idea of drinking to forget your troubles. That winter, I understood it. Your nerves are pulled tight all day, and a drink is the one thing that loosens them, so you keep reaching for it. Looking back, I think I was close to my limit. My body was telling me so.

Set up for failure

My friends at Amazon, most of them economists, watched all this with real concern. One after another, they told me to come back. I thanked them and said: let me wait a little longer.

I stayed because the reason I had come to Facebook was still there. I had come for product impact, and even that winter, my work was reaching the product every week, mistakes and all. I also wrote a long email to my skip-level manager, the person who had hired me, laying out what was happening and what I needed. My skip-level acted on it, and that support turned out to be one of the main reasons I was later able to turn things around. The turnaround came slowly, over 2020. That part of the story comes in Chapter 11.

Much later, back in Seattle, I had a conversation with that engineering manager, who by then had long since moved to another team. We talked about that first winter, and the engineering manager summed it up in one sentence: "Yeah, we were set up for failure back then."

I've never heard it put better. As I came to see it, that team was not going to succeed no matter what it did. Doing more might only have forced the people above to find a more strained good reason for a verdict that, I suspect, had already been reached.

At one point, after all of it, the same engineering manager built me a poker table by hand, as a thank-you for how hard I had worked during that period, and out of regret that I'd walked straight into that situation. We're still good friends. The scoreboard had called that season a failure. The person who worked through it beside me saw it differently.

Looking for the real reason

Now, when something goes wrong, I look for the real reason first. Before that winter, my first instinct was always to ask what I had done wrong.

Part of why it took me so long is that school had taught me the opposite. In an exam, the stated reason is the real one. If your score was low, your answers were wrong, and nobody needed a hidden motive to explain it. People like me, the ones who were top students all their lives, carry that habit into work. We are very good at finding the cause in ourselves. If the result was bad, we didn't work hard enough, or we weren't smart enough, or we should have seen it coming. There's a kind of pride in that. It means we're in control. But it also means that when we are in a bad environment, we keep grinding, telling ourselves it's our fault, when much of it isn't.

If I had seen the setup clearly that winter, I don't think I would have decided differently. But I would have suffered less. It was partly me: I lacked skills, and I lacked context. Much of it, though, was a scoreboard that couldn't tell our work from the

weather, used to supply a good reason for a decision made somewhere else.

Looking for the real reason doesn't mean assuming every verdict is political, or that every good reason is a lie. Often the good reason is the real one. It means asking, before you judge yourself, what the scoreboard could actually see, and who needed it to say what. That has a limit, though. One scoreboard isn't set by anyone above you. You keep it yourself.

Takeaway

Every verdict on your work comes with a good reason and a real reason. When a scoreboard can't tell your effort from luck, the two can come apart completely, and the number becomes the good reason for a decision made somewhere else. Don't take a verdict like that as the truth about your work, and don't spend your career producing good reasons instead of value. Ask what the scoreboard could actually see, and what the real reason was.

END OF THE FREE SAMPLE

Feels Like Progress

Part One ends with one more chapter, on the ten percent of work that matters and the busy days that hide it. Then the book turns to what AI will do to fake work, to where fake work comes from, and to what will last. Along the way: a sign outside a McDonald's, a middle-school robot contest, and a grocery business started in graduate school.

Get the whole book, in paperback or ebook, at

MAKEWHATLASTS.COM