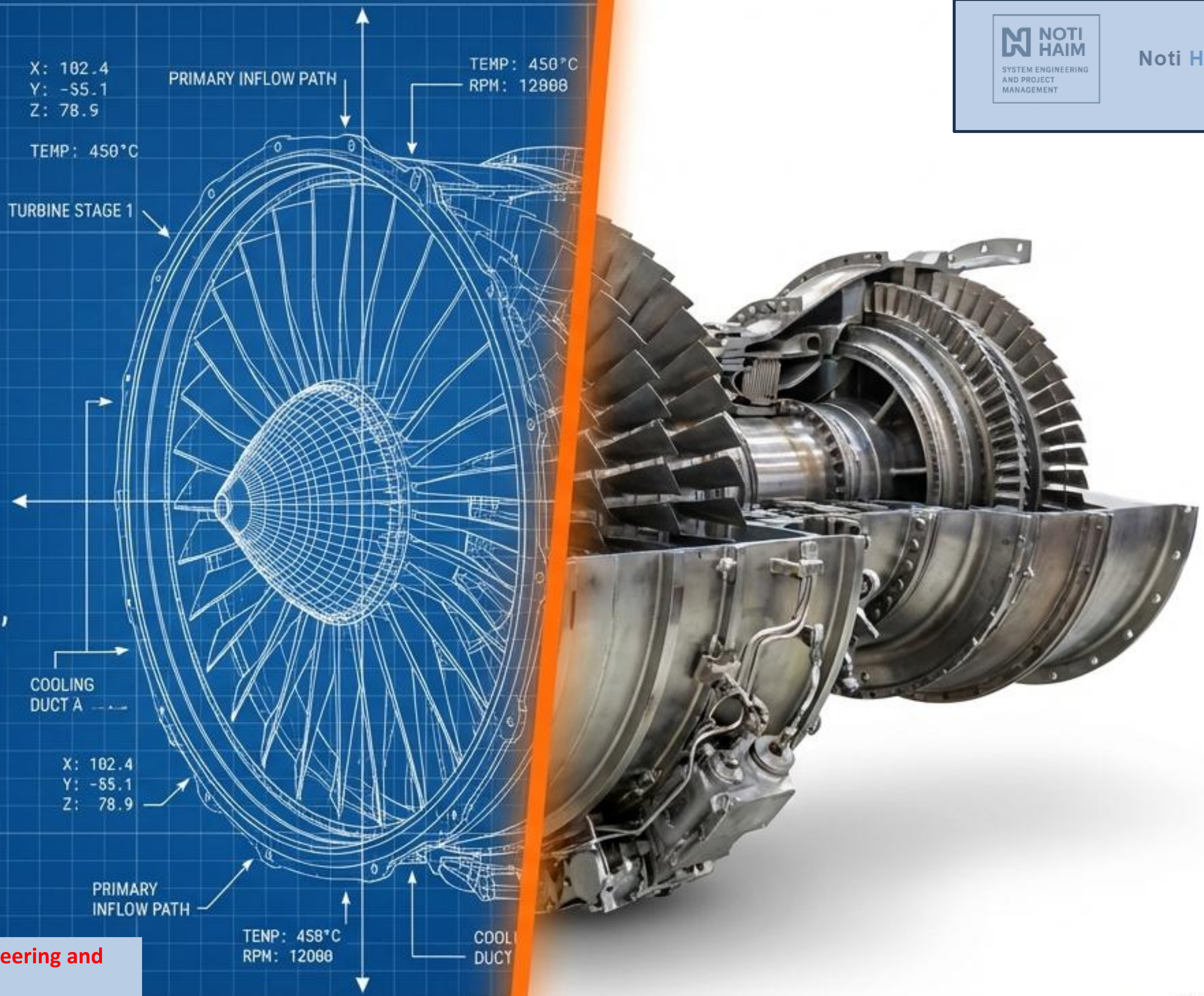
NOTI HAIM
SYSTEM ENGINEERING AND PROJECT MANAGEMENT

Noti Haim

Using Simulation: Is it Worth It?

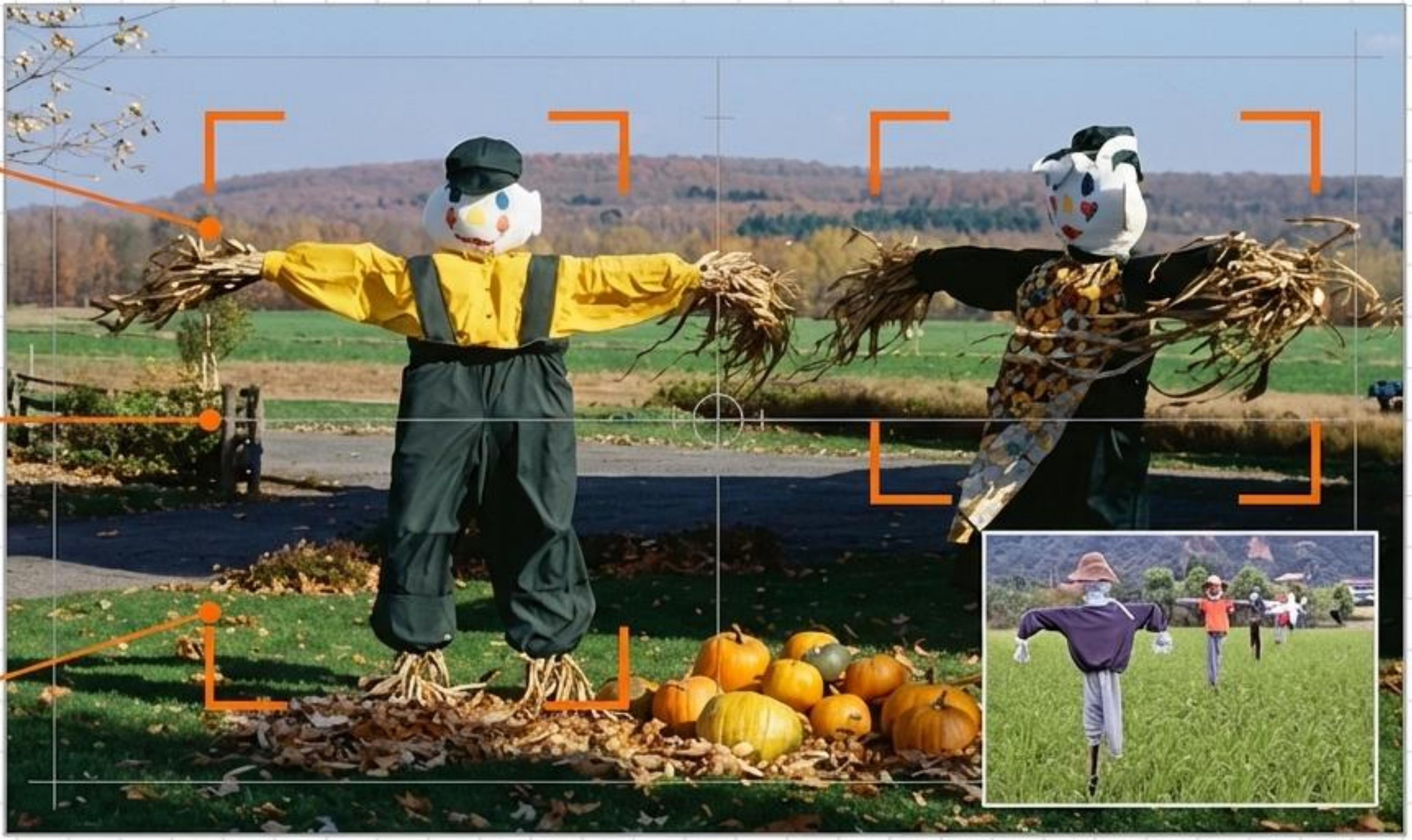
Navigating feasibility, cost, and risk in test engineering and training.



Property of Haim Noti – System Engineering and Project Management

NotebookLM

The Pragmatism of the Original Simulation



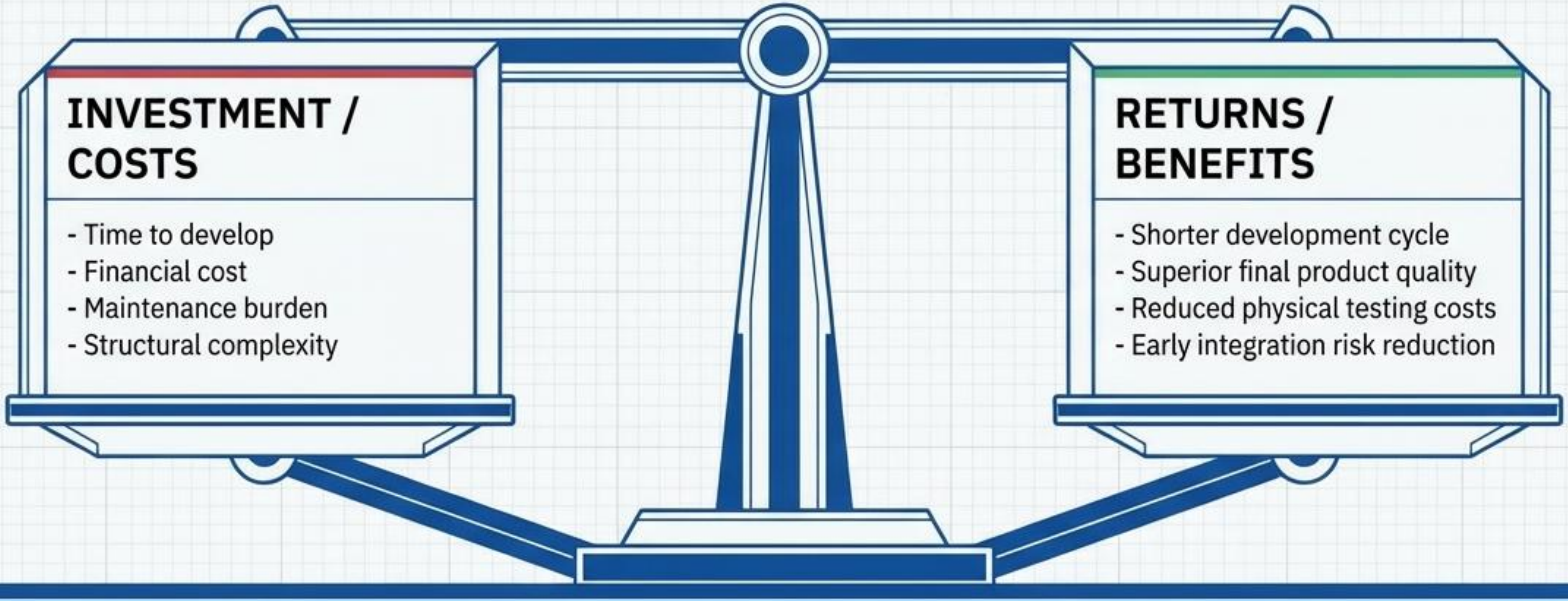
Low Fidelity: Does not biologically replicate a human.

High Efficacy: Perfectly models the relevant parameters to achieve the systemic outcome (scaring birds).

Low Resource Cost: Built from scrap; avoids deploying a real human to the field.

CORE INSIGHT: A simulation does not need to perfectly replicate reality. It only needs to model the exact parameters relevant to the engineering question being asked.

The Cost-Benefit Equation



INVESTMENT / COSTS

- Time to develop
- Financial cost
- Maintenance burden
- Structural complexity

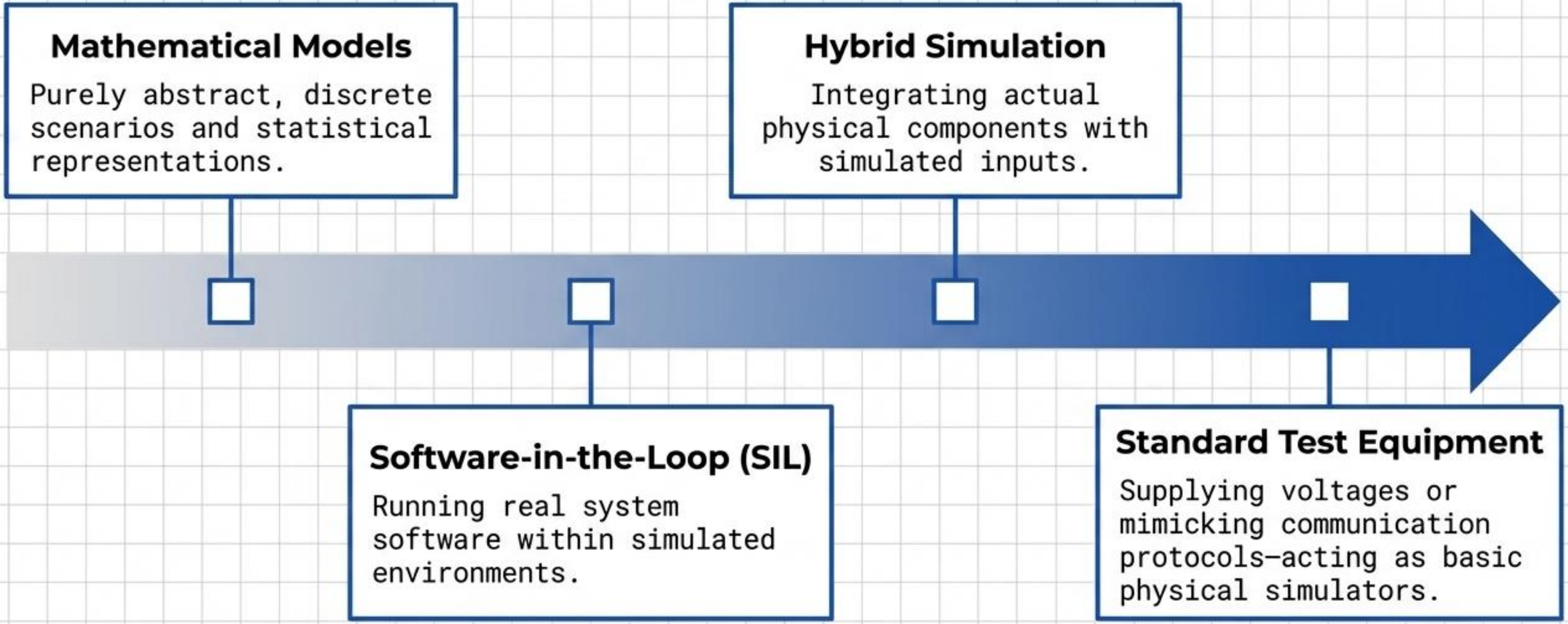
RETURNS / BENEFITS

- Shorter development cycle
- Superior final product quality
- Reduced physical testing costs
- Early integration risk reduction

A simulation is only viable if its cost in time and money is demonstrably smaller than the alternative physical testing routes. It must accelerate the development process, not become an isolated, bloated project of its own.

NotebookLM

Defining the Simulation Spectrum



Mathematical Models

Purely abstract, discrete scenarios and statistical representations.

Software-in-the-Loop (SIL)

Running real system software within simulated environments.

Hybrid Simulation

Integrating actual physical components with simulated inputs.

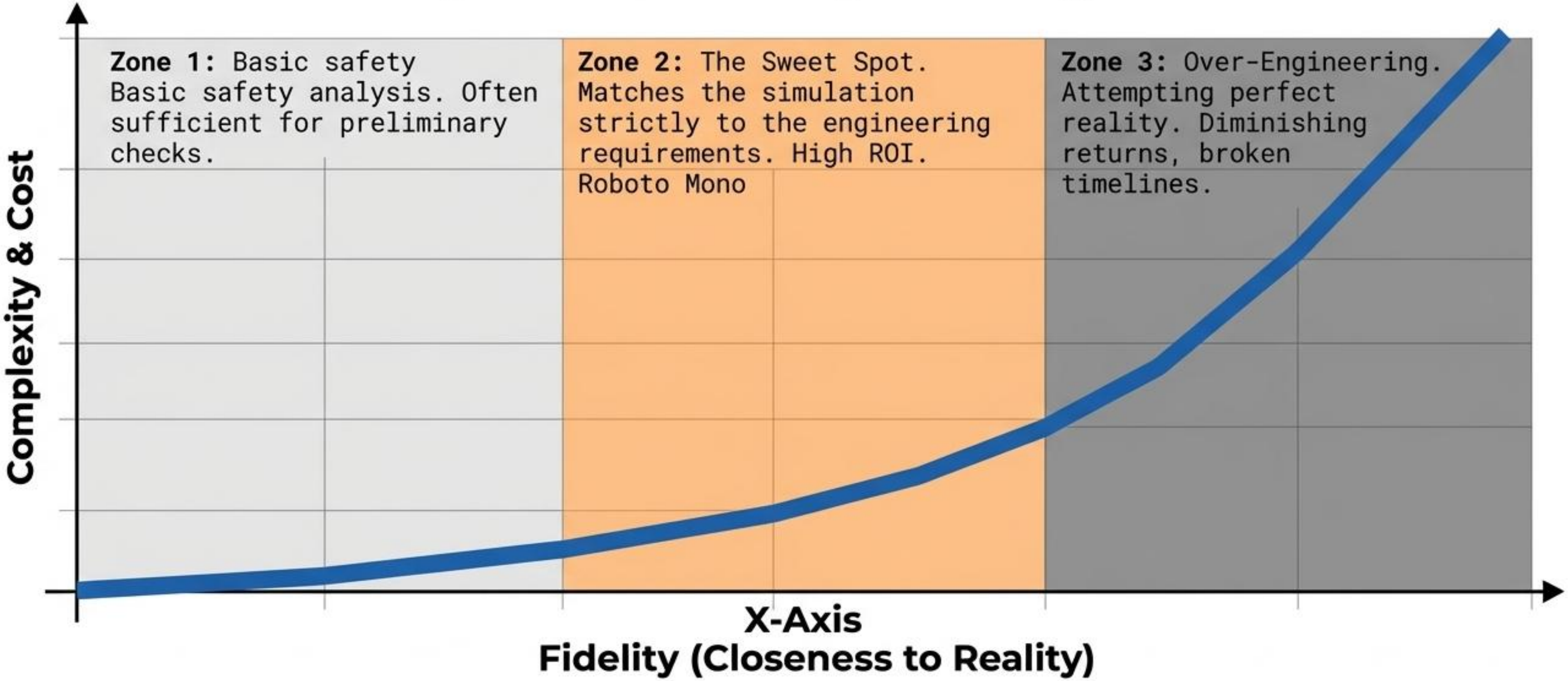
Standard Test Equipment

Supplying voltages or mimicking communication protocols—acting as basic physical simulators.

TAKEAWAY: Simulation is simply the modeling of a system's behavior—whether capturing a single parameter or the full dynamic scope.

NotebookLM

The Fidelity vs. Complexity Trade-Off



Zone 1: Basic safety
Basic safety analysis. Often sufficient for preliminary checks.

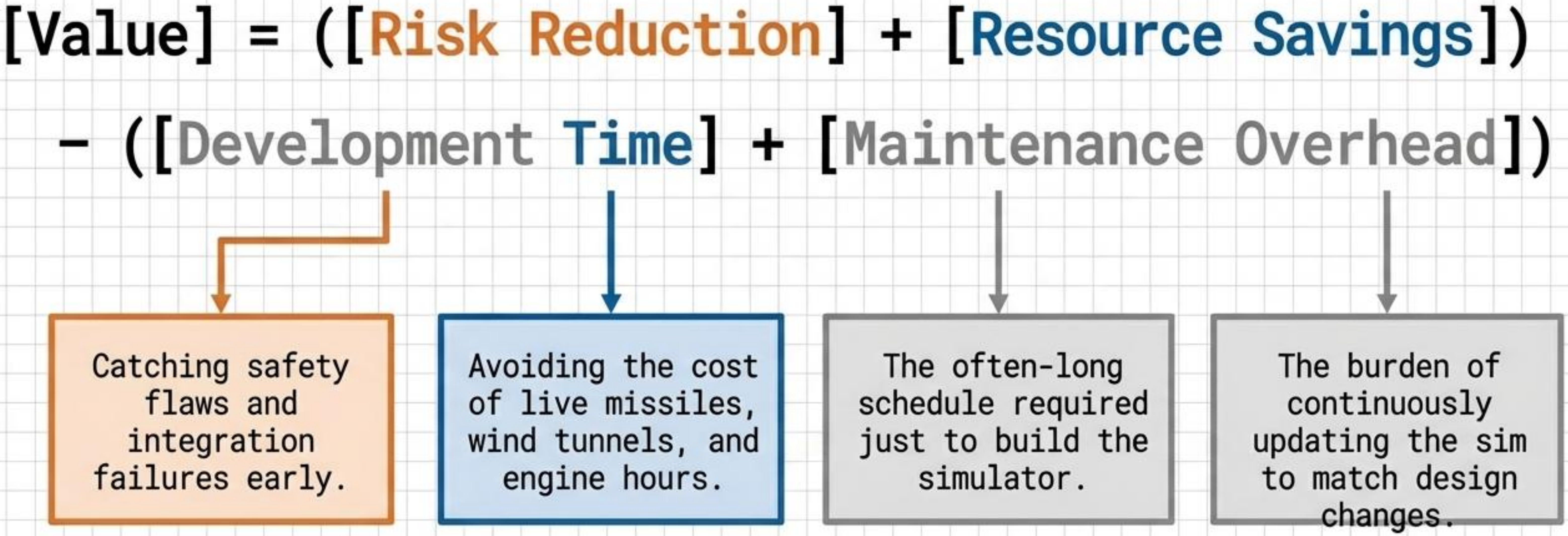
Zone 2: The Sweet Spot.
Matches the simulation strictly to the engineering requirements. High ROI.
Roboto Mono

Zone 3: Over-Engineering.
Attempting perfect reality. Diminishing returns, broken timelines.

Key Insight: Determine complexity strictly based on the specific engineering questions that require answers—not a desire for aesthetic realism.

NotebookLM

The Feasibility Equation: Is It Worth It?



[Value] = ([Risk Reduction] + [Resource Savings])

− ([Development Time] + [Maintenance Overhead])

Catching safety flaws and integration failures early.

Avoiding the cost of live missiles, wind tunnels, and engine hours.

The often-long schedule required just to build the simulator.

The burden of continuously updating the sim to match design changes.

THE RULE: The cost in time and money MUST be strictly smaller than the alternative physical testing and integration methods.

NotebookLM

Halting the Physical Burn Rate

Physical Reality Cost

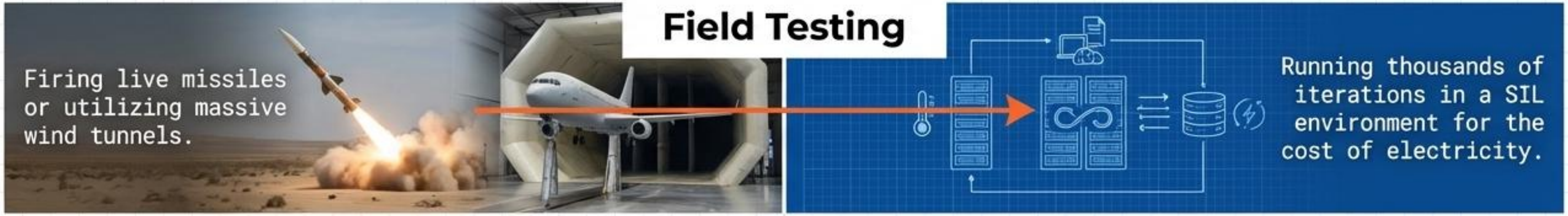
Simulated Model Cost

User Training

Expending invaluable “Engine Hours” on real machinery.

Infinite reset loops, preserving physical asset lifespans.

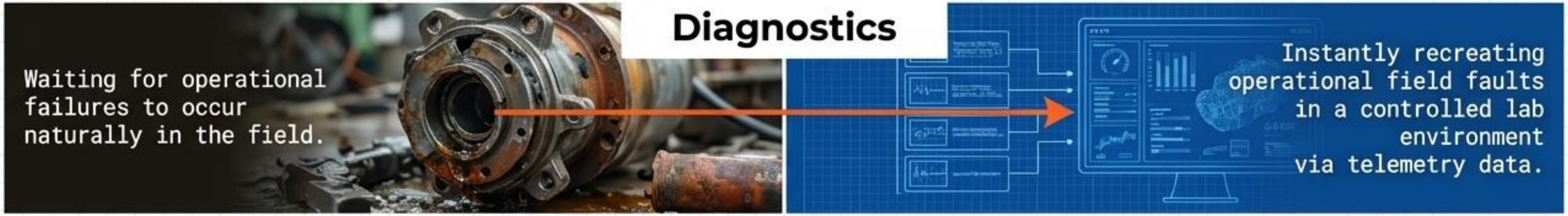
Field Testing



Firing live missiles or utilizing massive wind tunnels.

Running thousands of iterations in a SIL environment for the cost of electricity.

Diagnostics

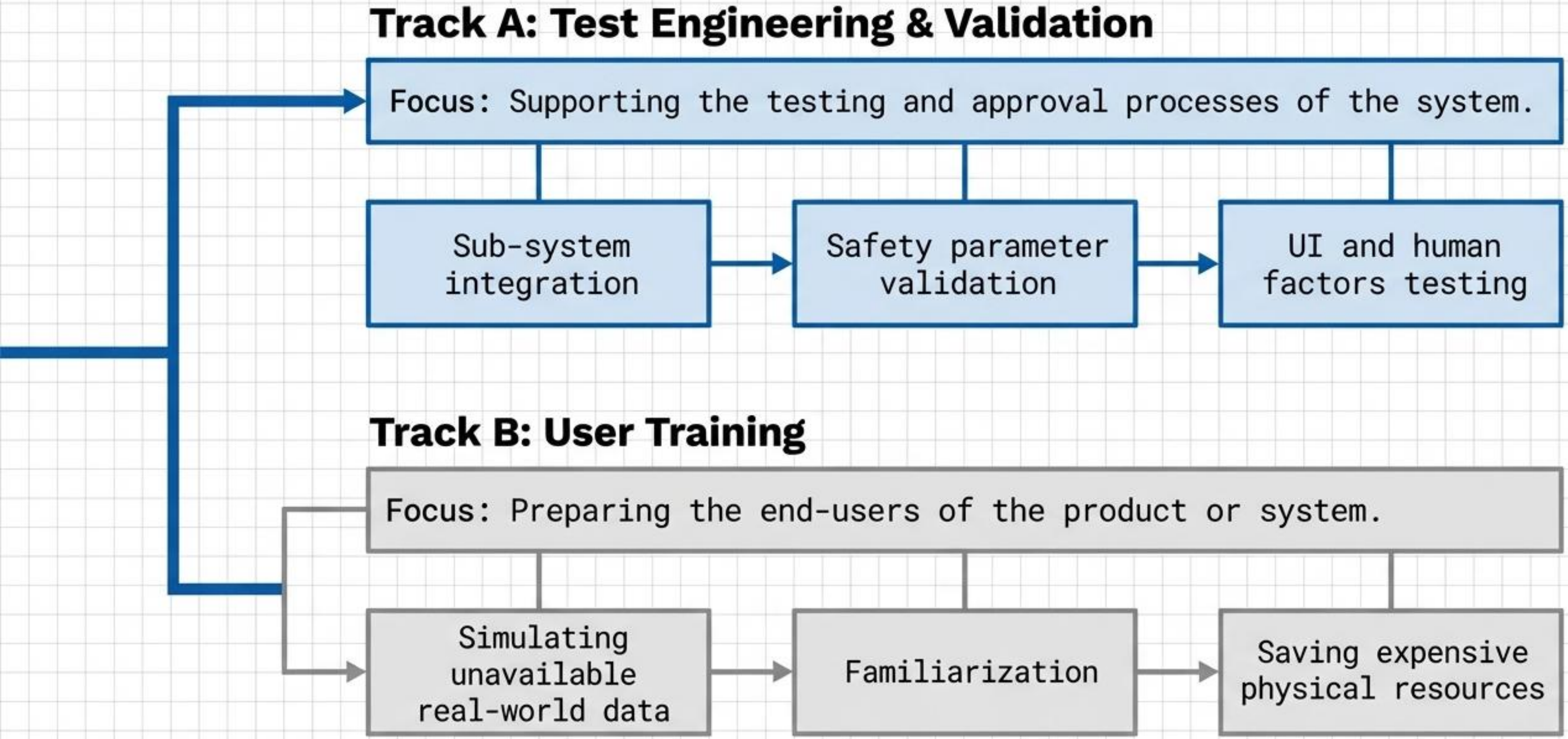


Waiting for operational failures to occur naturally in the field.

Instantly recreating operational field faults in a controlled lab environment via telemetry data.

NotebookLM

The Two Distinct Paths of Simulation Adoption



Track A: Test Engineering & Validation

Focus: Supporting the testing and approval processes of the system.

Sub-system integration → Safety parameter validation → UI and human factors testing

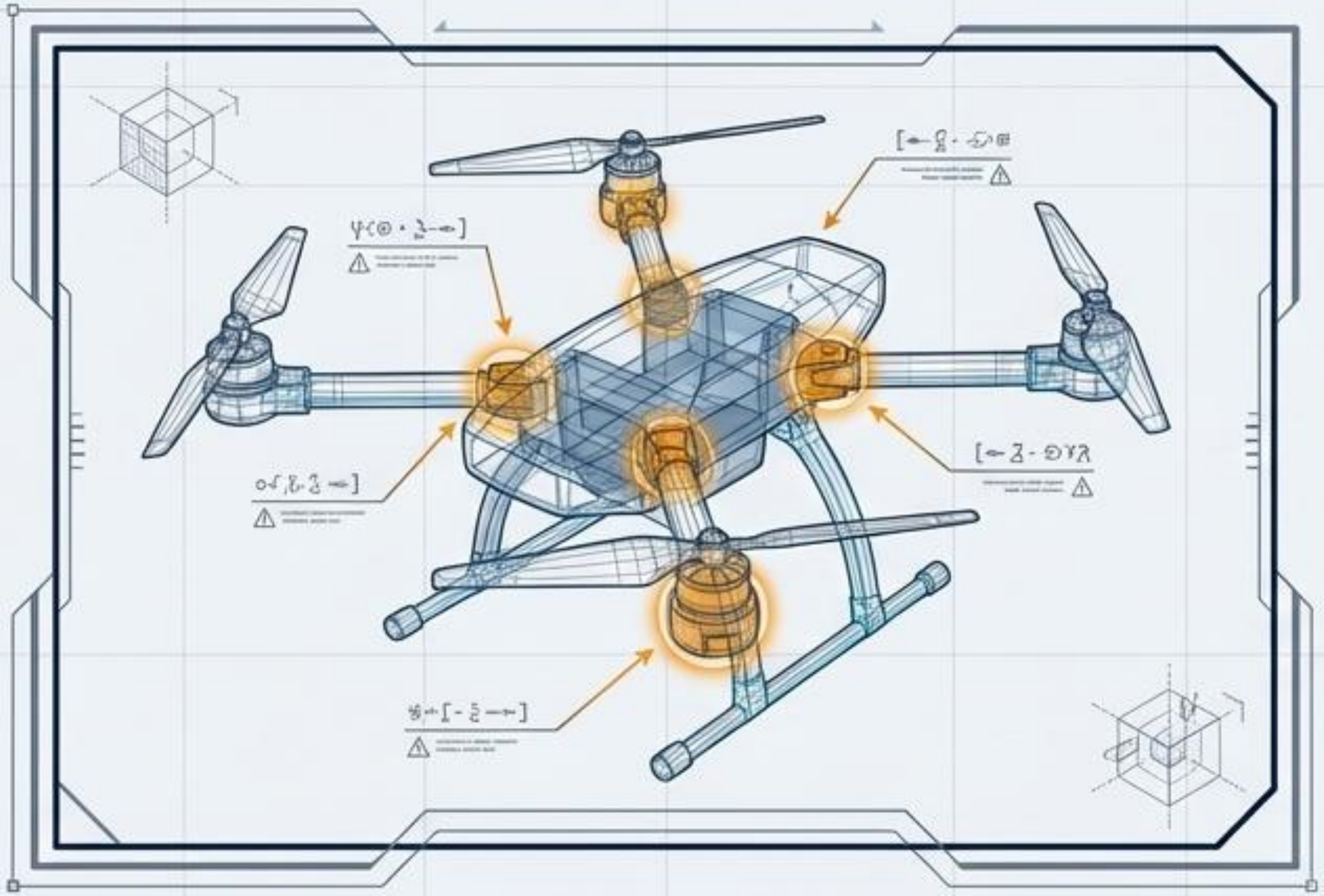
Track B: User Training

Focus: Preparing the end-users of the product or system.

Simulating unavailable real-world data → Familiarization → Saving expensive physical resources

NotebookLM

Pillars 1 & 2: Design Support & Parameter Tuning

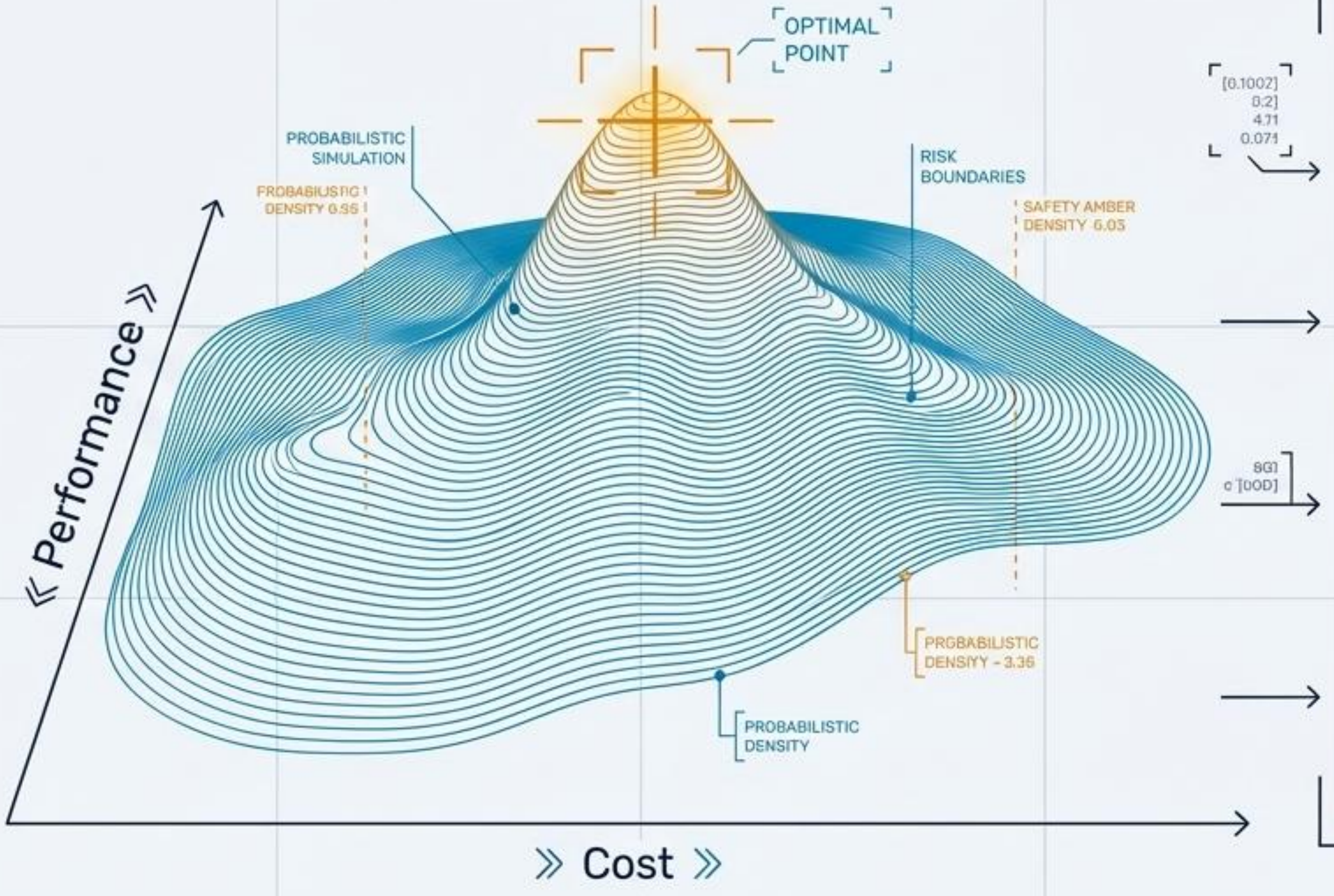


Design Support / תמיכה בתכן:
- Architecture evaluation & Trade-offs.
- Early risk detection / הורדת סיכונים בשלב מוקדם.
- Performance estimation.

Parameter Tuning / כיול פרמטרים:
- Controller gains (PID).
- Threshold tuning & Model calibration.
- Accuracy improvement / פרמטרי כיול.

NotebookLM

Pillar 3: Algorithm Optimization



Search optimal parameters

כיוון ואופטימיזציה לנקודת העבודה של המערכת.

Monte Carlo analysis

Running thousands of scenarios to find probabilistic edge cases.

Performance vs Cost

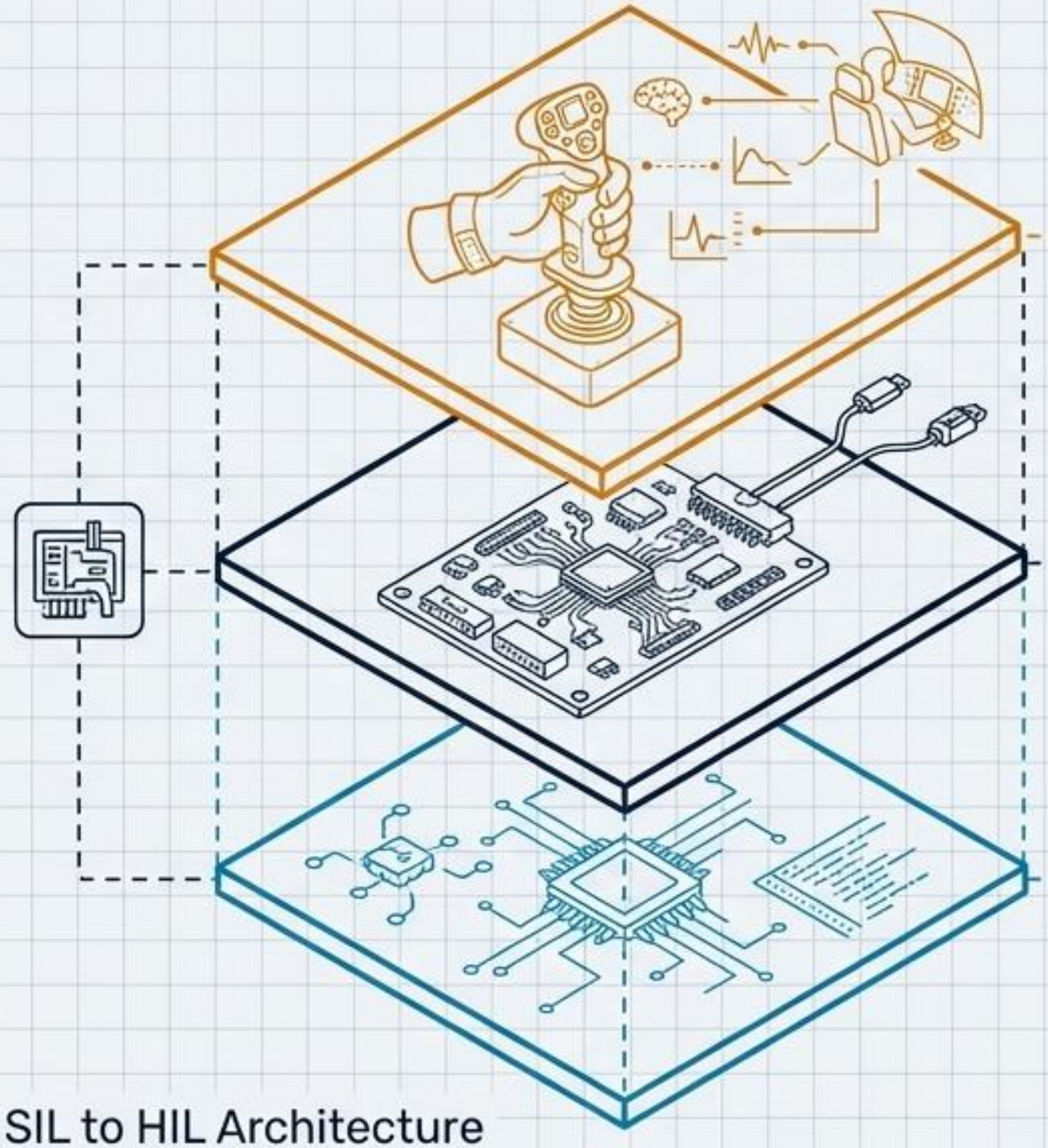
Balancing efficiency with reality trade-offs.

Robustness & Sensitivity

חקר ביצועים, סטטיסטיקה, רגישות, רובוסטיות.

NotebookLM

Pillar 4: Verification & Validation (אימות ותיקוף)



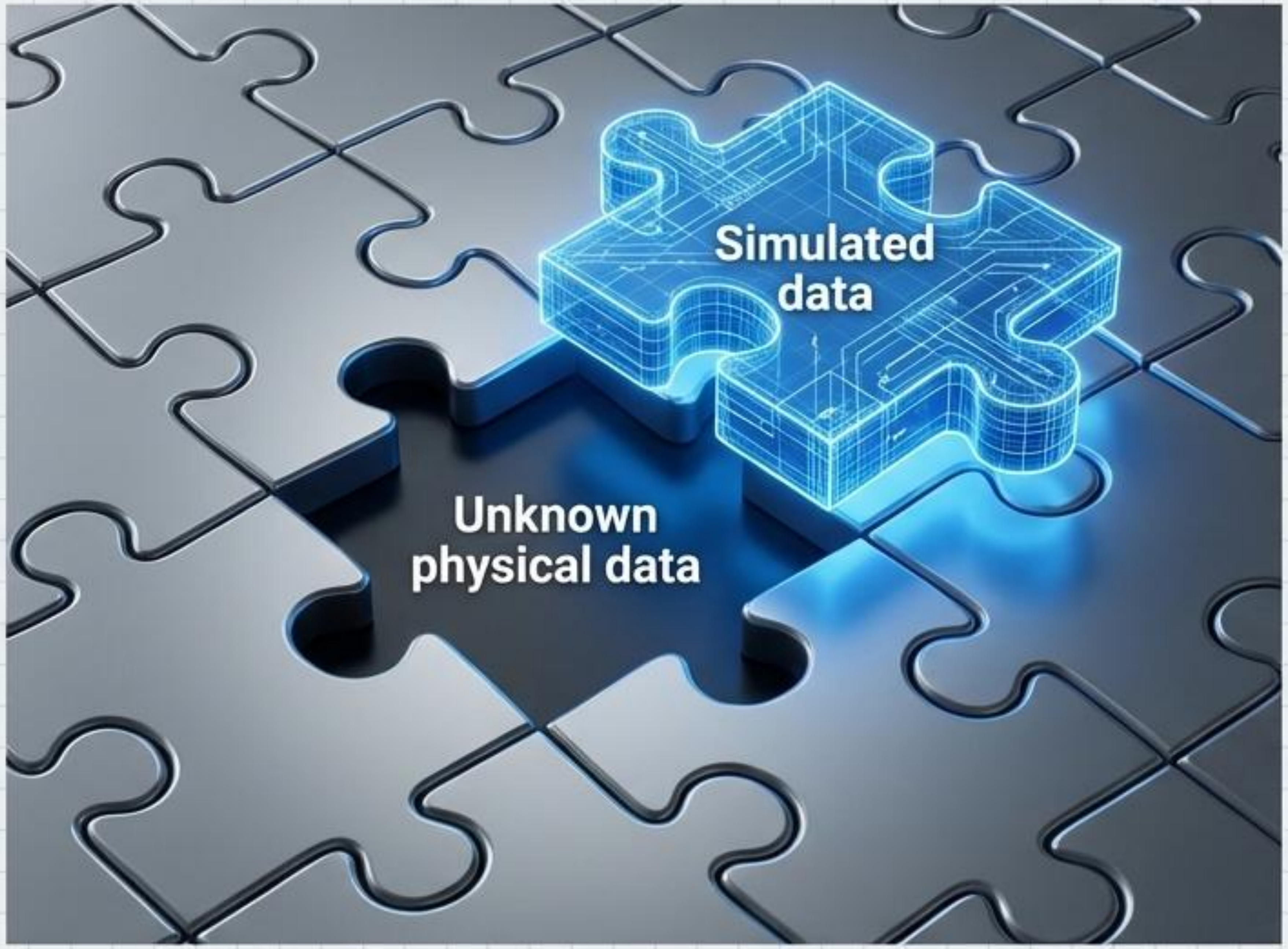
- **HITL (Human-in-the-Loop)**
  The final frontier integrating the human operator and physiological latency.
- **HIL (Hardware-in-the-Loop)**
  Code integrated with physical COTS and custom hardware components.
- **SIL (Software-in-the-Loop)**
  Pure code and algorithmic testing.

SIL to HIL Architecture

- Testing edge cases that are too dangerous for physical trials.
- Requirement validation against real-world comparisons.

NotebookLM

Testing the Unknown



Often, system performance must be validated before real-world data or final hardware is available.

Key Mechanisms:

Synthetic Generation: Generating simulated data to feed into algorithms.

Sensitivity Checks: Testing how the system reacts across a wide band of unknown parameters.

Calibration Integration: Utilizing complex calibration parameters inside the simulation to tune the system until physical data can validate the model.

The To Sim or Not To Sim Diagnostic Table

Scenario	Recommended Approach	Key Advantage	Primary Risk
Complex systems with expensive physical trials (e.g., Missiles, Wind Tunnels).	Heavily Simulate.	Massive cost/resource savings.	Complex, long development time.
High safety risks prior to integration.	Simulate (if analysis fails).	Lowers pre-integration danger.	Standard safety analysis might have sufficed.
Software-heavy systems utilizing COTS hardware.	Heavily Simulate.	Accelerates algorithmic development.	Hardware nuances missed.
Standard production line and routine maintenance testing.	Do Not Simulate.	Direct physical validation.	Requires building separate, dedicated test equipment.

NotebookLM

Where Simulation Outperforms Reality

Extreme Edge Cases & Safety

Testing system limits outside the normal operational envelope (e.g., weapon release dynamics, target missile pressure, high-humidity video performance) safely without risking physical prototypes or human life.

Resource Conservation

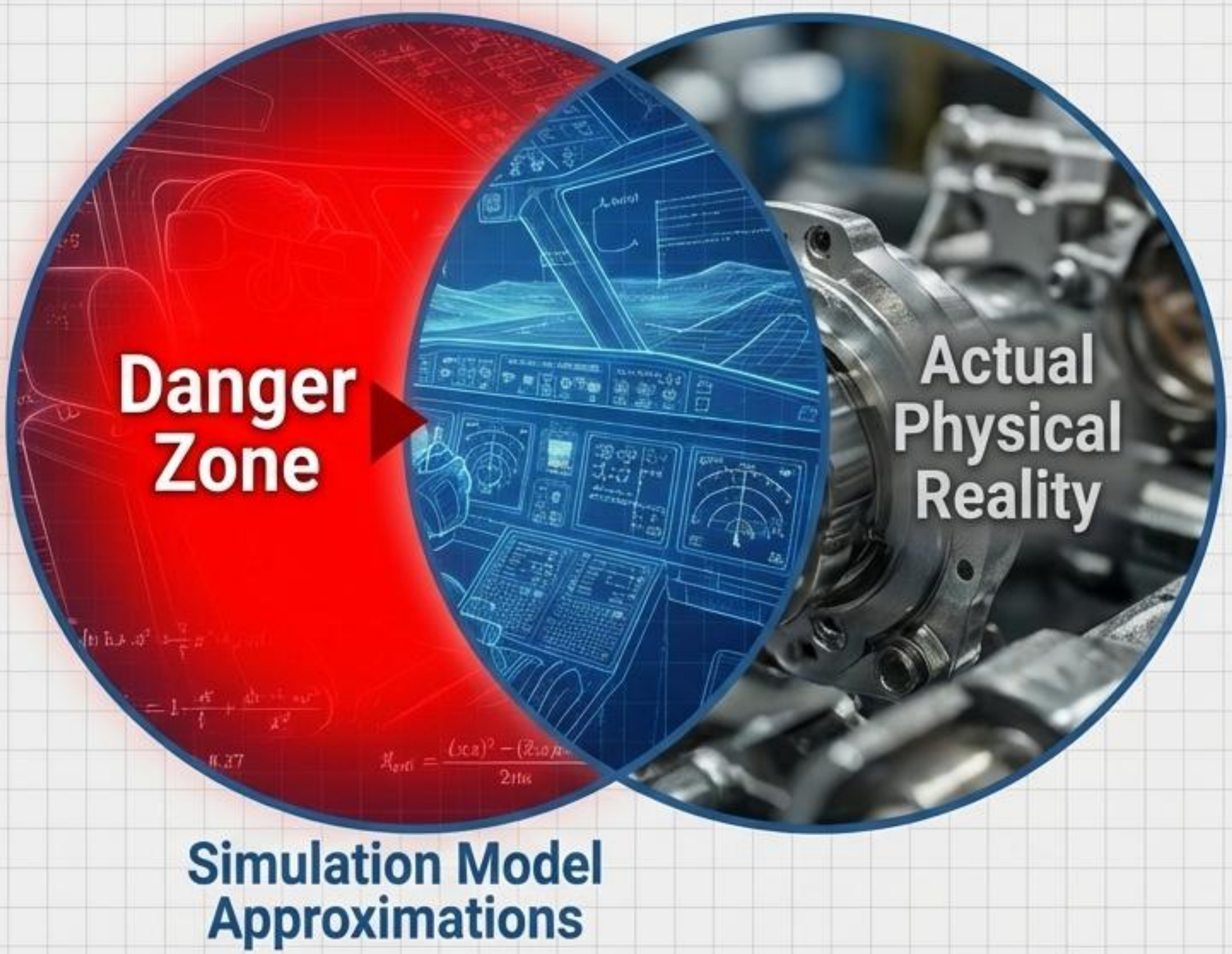
Replacing highly expensive physical trials. Bypassing the need to burn expensive engine hours, rent wind tunnels, or fire live missiles for early-stage validation.

Early Customer Integration

Testing against the final environment before the final product exists. Example: Performing Software-in-the-Loop (SIL) flight simulations long before the physical aircraft integration phase.

NotebookLM

The Hidden Risks of Simulation



The False Confidence Trap: Reaching flawed engineering conclusions because the simulation relies on incorrect approximations or bad math.

Schedule Bloat: The simulator development timeframe stretches, failing to align with the actual product development schedule.

The Maintenance Burden: The massive, continuous effort required to update the simulation every time the physical product design matures or changes.

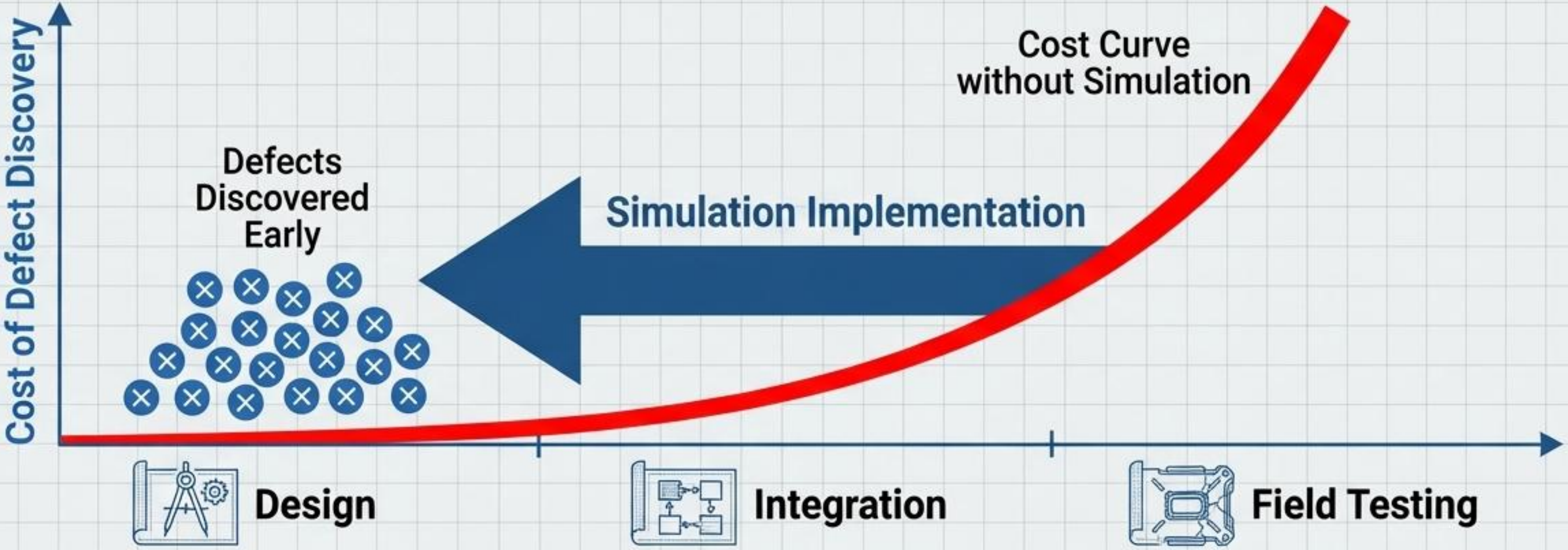
NotebookLM

Human-in-the-Loop (HITL) Deep Dive

Dimension	Pure Machine Simulation	Human-Integrated Reality
Reaction Time	Near-Zero Latency	Startle Delay / 35s Latency
Decision Process	Binary / Algorithmic Execution	Cognitive Load & Ambiguity Processing
System Impact	Predictable & Perfect Execution	Erratic initially, but adaptable to unprogrammed edge cases

Never optimize the human out of the loop. Real systems include decision latency.

Shifting Risk to the Left



Key Insight

A critical feasibility parameter is the immense risk of discovering architectural flaws during late-stage physical integration.

Takeaway

Simulation mitigates the massive cost of late-stage system changes by forcing integration issues to surface mathematically before metal is ever cut.

NotebookLM

The Capability Matrix: Sim vs. Physical

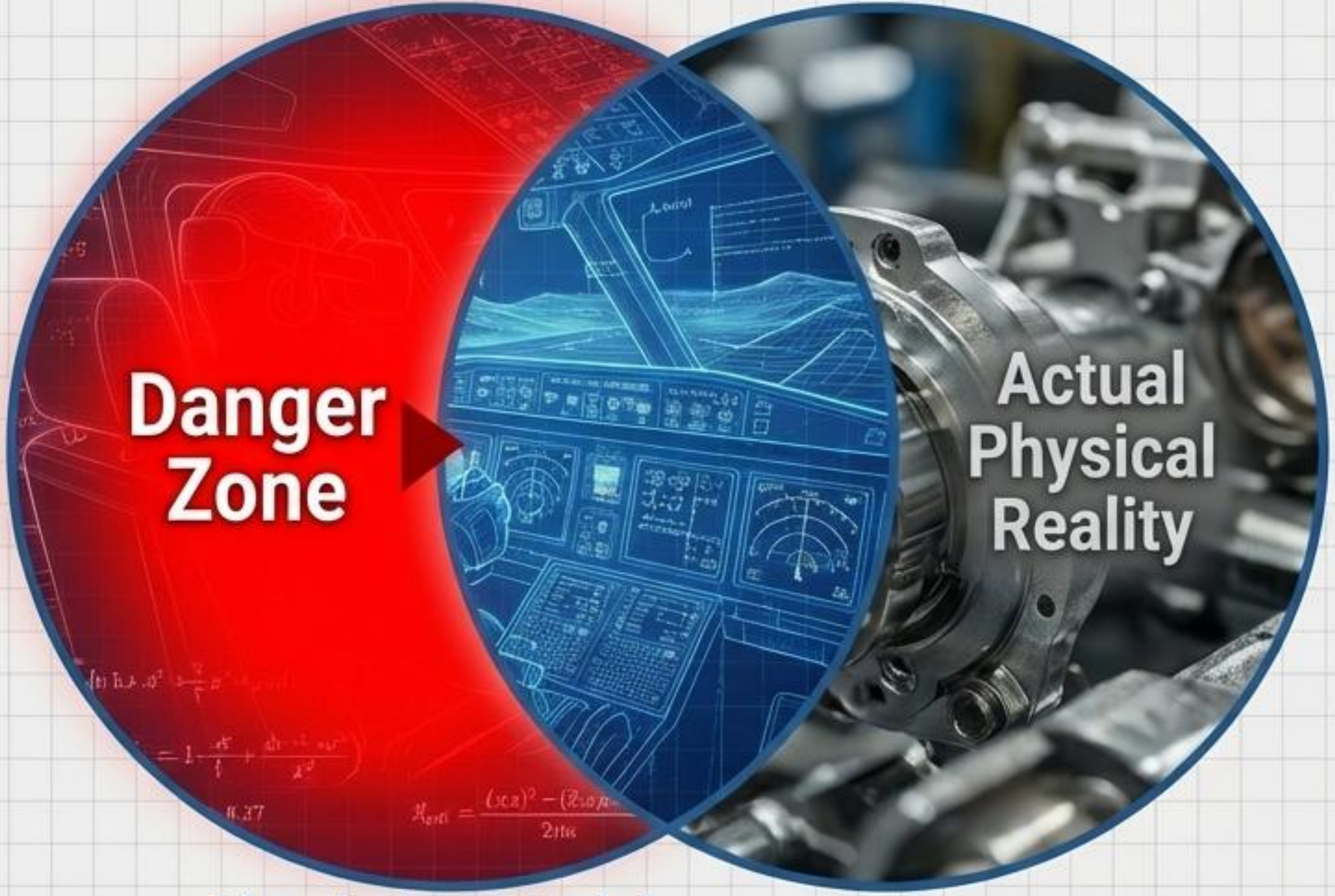
DIN Next LT Pro	Simulation	Physical Testing
Extreme Edge-Case Safety	●	○
Initial Setup Cost & Time	○	○
Resource Consumption (Engine hours, hardware)	●	○
Early-Stage Availability	●	○
Absolute Physical Reality (Ground Truth)	○	●

Data Highlight

Simulation wins heavily on Safety, Resource Consumption, and Early Availability. Physical Testing wins exclusively on Ground Truth and often requires dedicated production-line test equipment that simulators cannot replace.

NotebookLM

The Hidden Risks of Simulation



Simulation Model Approximations

The False Confidence Trap: Reaching flawed engineering conclusions because the simulation relies on incorrect approximations or bad math.

Schedule Bloat: The simulator development timeframe stretches, failing to align with the actual product development schedule.

The Maintenance Burden: The massive, continuous effort required to update the simulation every time the physical product design matures or changes.

The To Sim or Not To Sim Diagnostic Table

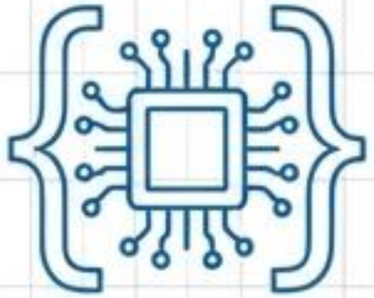
Scenario	Recommended Approach	Key Advantage	Primary Risk
Complex systems with expensive physical trials (e.g., Missiles, Wind Tunnels).	Heavily Simulate.	Massive cost/resource savings.	Complex, long development time.
High safety risks prior to integration.	Simulate (if analysis fails).	Lowers pre-integration danger.	Standard safety analysis might have sufficed.
Software-heavy systems utilizing COTS hardware.	Heavily Simulate.	Accelerates algorithmic development.	Hardware nuances missed.
Standard production line and routine maintenance testing.	Do Not Simulate.	Direct physical validation.	Requires building separate, dedicated test equipment.

NotebookLM

The Case for YES: Where Simulation Dominates

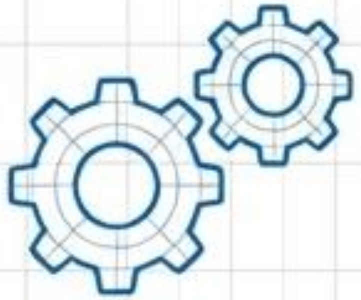
Complex & Expensive Systems

Ideal for replacing highly expensive physical assets (e.g., live missiles, wind tunnels).



Algorithm & Software Heavy

Especially valuable when the hardware relies heavily on COTS (Commercial Off-The-Shelf) components.



Concurrent Client Development

Crucial for early risk reduction when the final client system is not yet physically available for integration.

Investigating the Unknown

Operations research, statistics, system sensitivity, and pinpointing the optimal working point before manufacturing.

NotebookLM

The Case for NO: The Simulation Trap

The Easy-to-Field System

Red Flag: The system is relatively simple to integrate and test outdoors.

Reality: Building a sim costs more time and money than simply taking the prototype outside.

Custom Hardware Dependency

Red Flag: The system is deeply reliant on unique, proprietary hardware.

Reality: Simulating custom hardware is extraordinarily difficult and typically requires separate, dedicated manufacturing/maintenance test gear anyway.

The Schedule Breaker

Red Flag: The time required to develop the simulator exceeds the product development timeline.

Reality: A delayed simulator halts the master schedule, negating any potential speed benefits.

Diagnostic Matrix: Selecting the Right Validation Tool

	Live Field Testing	Simulation / SIL	Safety Analysis
System Complexity	Low	High	Low-Med
Safety Risk	Low	Extreme	Moderate
Primary Asset Cost	Cheap/Easy	Expensive/Missiles	N/A
System Type	Hardware-heavy	Software/COTS-heavy	Logic-based
Real System Availability	Available	Concurrent Dev/Unavailable	In Design

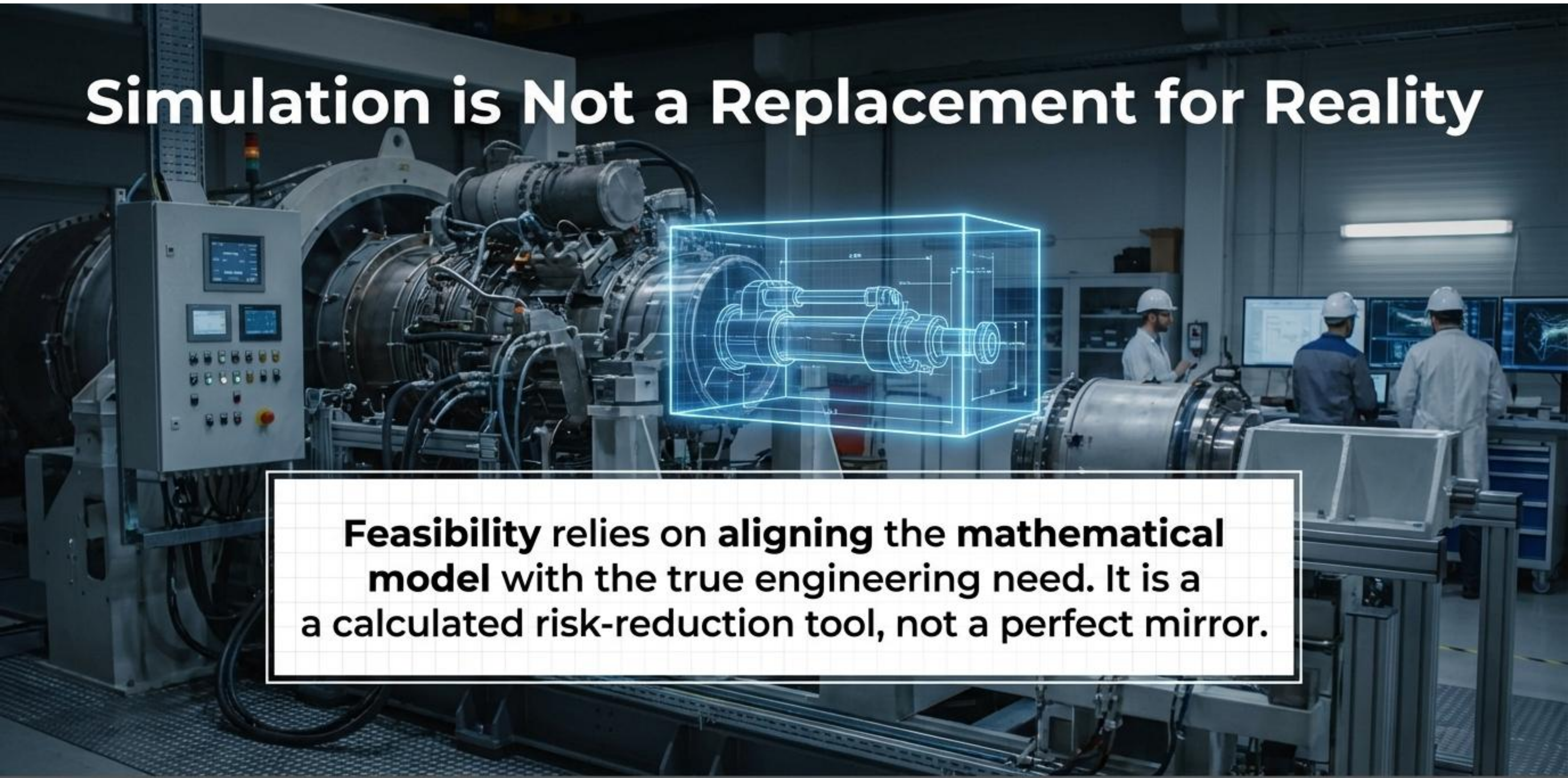
NotebookLM

The “Worth It” Go / No-Go Checklist

- ☐ **Schedule Alignment:** Does the simulator development time fit seamlessly within the master project schedule?
- ☐ **Economic Viability:** Is the cost of building the sim demonstrably lower than the physical tests it replaces?
- ☐ **Update Capacity:** Are resources allocated to continuously update the sim as the physical system evolves?
- ☐ **Component Focus:** Are we modeling only the relevant requirements, resisting the urge to over-engineer fidelity?
- ☐ **Training Adequacy:** Can the simulation genuinely provide adequate user training, or is a real system strictly required?

ACTION: If any box remains unchecked, default to physical integration or basic safety analysis.

NotebookLM



Simulation is Not a Replacement for Reality

Feasibility relies on **aligning** the **mathematical model** with the true engineering need. It is a a calculated risk-reduction tool, not a perfect mirror.

Ultimately, the viability of a system-level simulation is judged entirely against its ability to meet the schedule, reduce danger, and prove its economic worth.

NotebookLM

NOTI HAIM
SYSTEM ENGINEERING AND PROJECT MANAGEMENT

Noti Haim

Sully: Simulation vs. Reality

The Human-in-the-Loop Gap in Flight 1549

A systemic teardown of simulation assumptions versus operational reality.

Property of Haim Noti – System Engineering and Project Management

NotebookLM

The Mathematics of an Ideal Return

- The NTSB simulations made a damning initial claim: Under ideal conditions, the aircraft could have successfully returned to an airport runway.
- The math worked perfectly. The algorithms mapped a survivable trajectory.
- According to the machine, the Hudson River ditching was an unnecessary and reckless risk.

SIMULATION A
(LaGuardia)

SIMULATION B
(Teterboro)

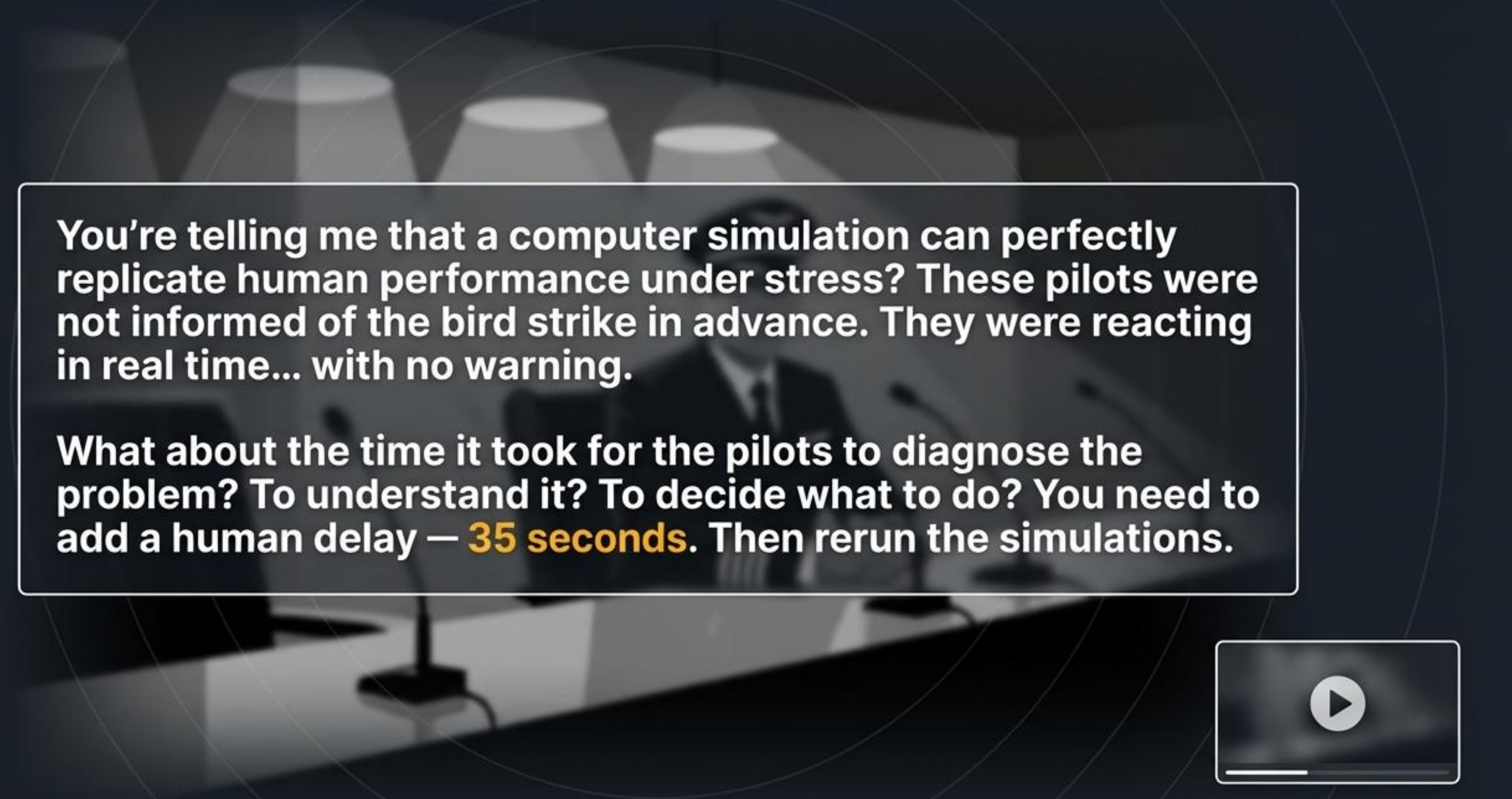
The Flaw in the Machine

Mapping the Model vs. Reality Gap

	Ideal System Simulation	Operational Reality
Reaction Time	0.0s (Instantaneous execution)	Human Latency
Prior Knowledge	Pre-warned of exact fault	Total ambush / Complete surprise
Cognitive Load	Zero (Algorithm executing logic)	Extreme trauma and uncertainty
System Fidelity	Machine-only parameters	Human-in-the-loop dependencies

NotebookLM

Can a Computer Replicate Human Performance Under Stress?



You’re telling me that a computer simulation can perfectly replicate human performance under stress? These pilots were not informed of the bird strike in advance. They were reacting in real time… with no warning.

What about the time it took for the pilots to diagnose the problem? To understand it? To decide what to do? You need to add a human delay — 35 seconds. Then rerun the simulations.

Watch the Hearing: https://www.youtube.com/watch?v=792tcJQTpN0

NotebookLM

The Physics of Decision Latency

00:00 - 00:12

Recognition:
Processing unexpected sensory input ("What was that sound and shudder?")

00:12 - 00:24

Understanding:
Diagnosing complete system failure without pre-warning ("We have lost thrust in both engines.")

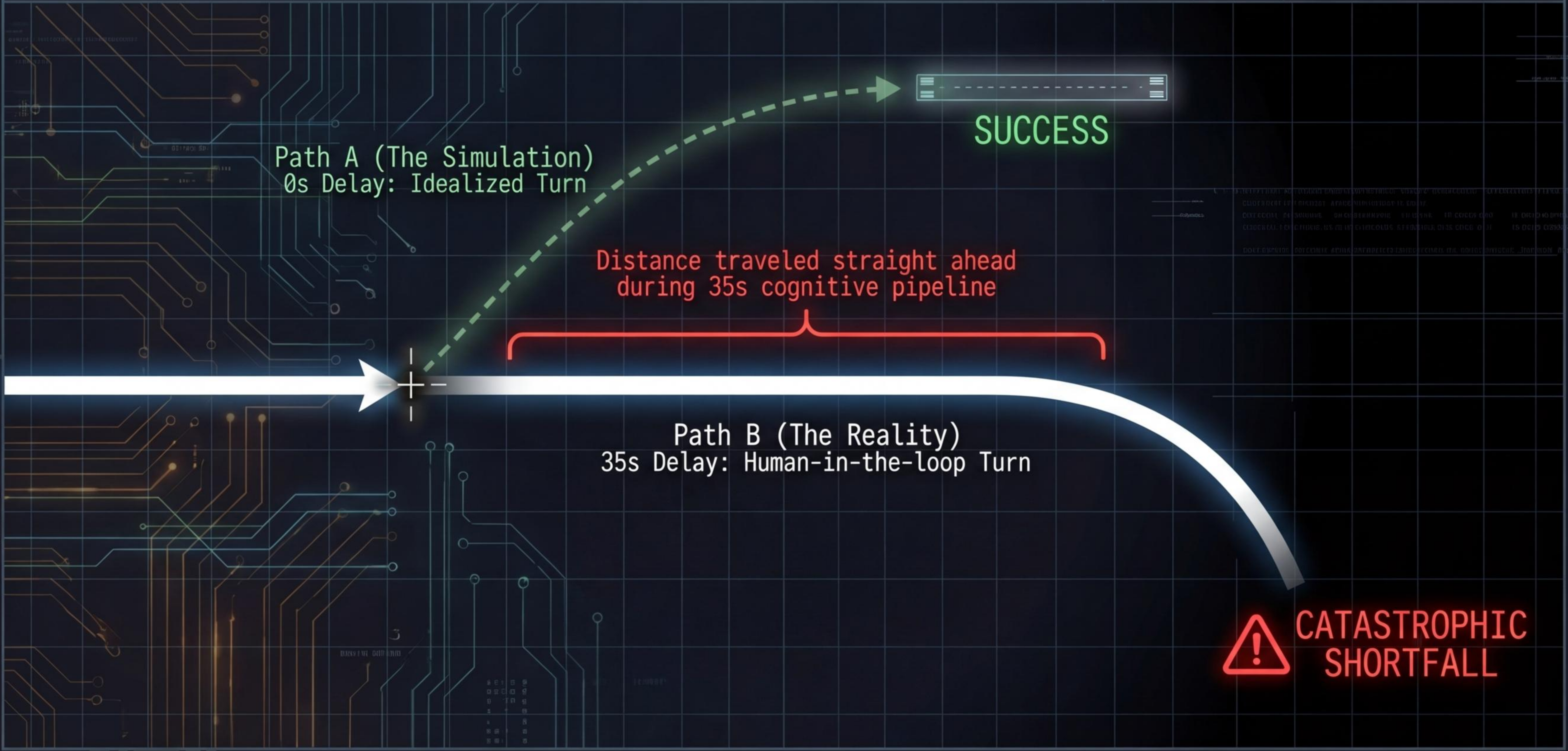
00:24 - 00:35

Decision:
Calculating real-world constraints and committing to a totally novel solution ("We cannot reach a runway; we must ditch.")

This **35-second** human-in-the-loop delay is an absolute operational constant. It cannot be engineered away.

NotebookLM

The spatial cost of human decision latency



Path A (The Simulation)
0s Delay: Idealized Turn

SUCCESS

Distance traveled straight ahead
during 35s cognitive pipeline

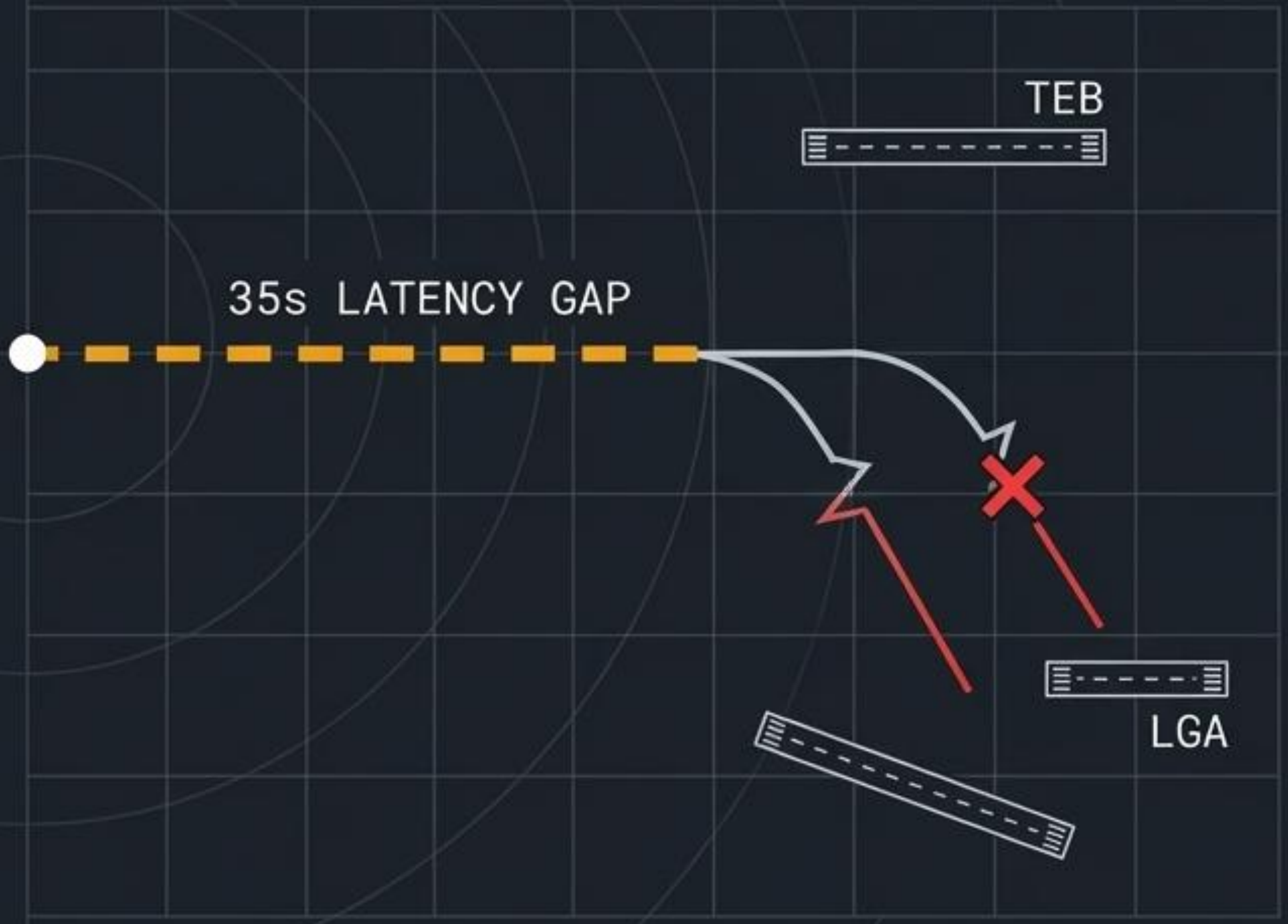
Path B (The Reality)
35s Delay: Human-in-the-loop Turn

CATASTROPHIC
SHORTFALL

NotebookLM

Introducing Human Latency Breaks the Model

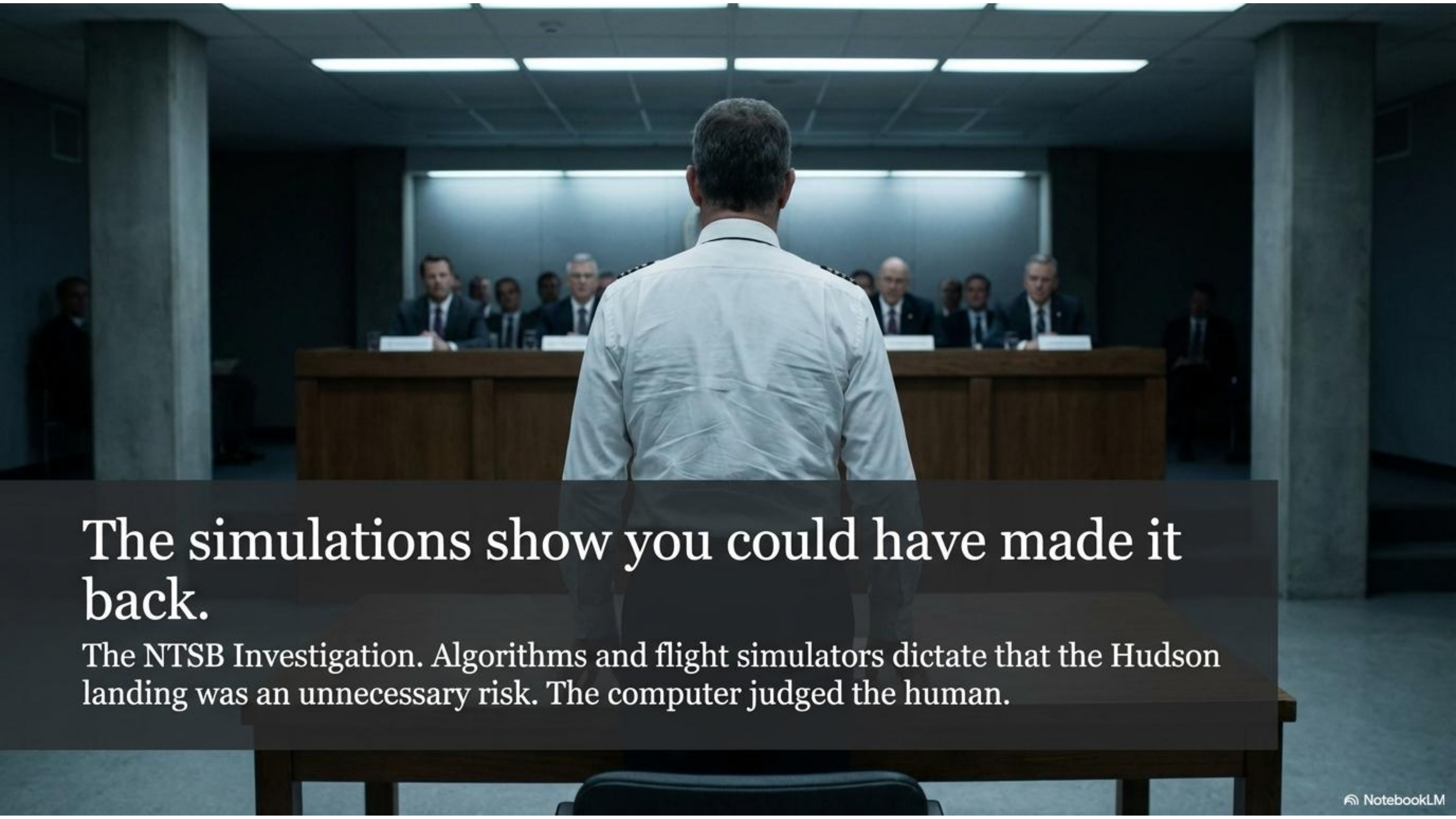
- When 35 seconds of diagnosis and decision delay are factored into the physics engine, the airport returns fail 100% of the time.
- The simulation proves that under operational reality, the Hudson landing was the only mathematically and humanly viable option.



The Model vs. Reality Matrix

Simulation Assumptions	Real-World Constraints
Instantaneous execution (Zero latency)	Human-in-the-loop delay (35 seconds)
Perfect situational awareness	High operational uncertainty
Machine-level processing speed	Overwhelming cognitive load
Static environmental variables	Dynamic, degrading physical environment

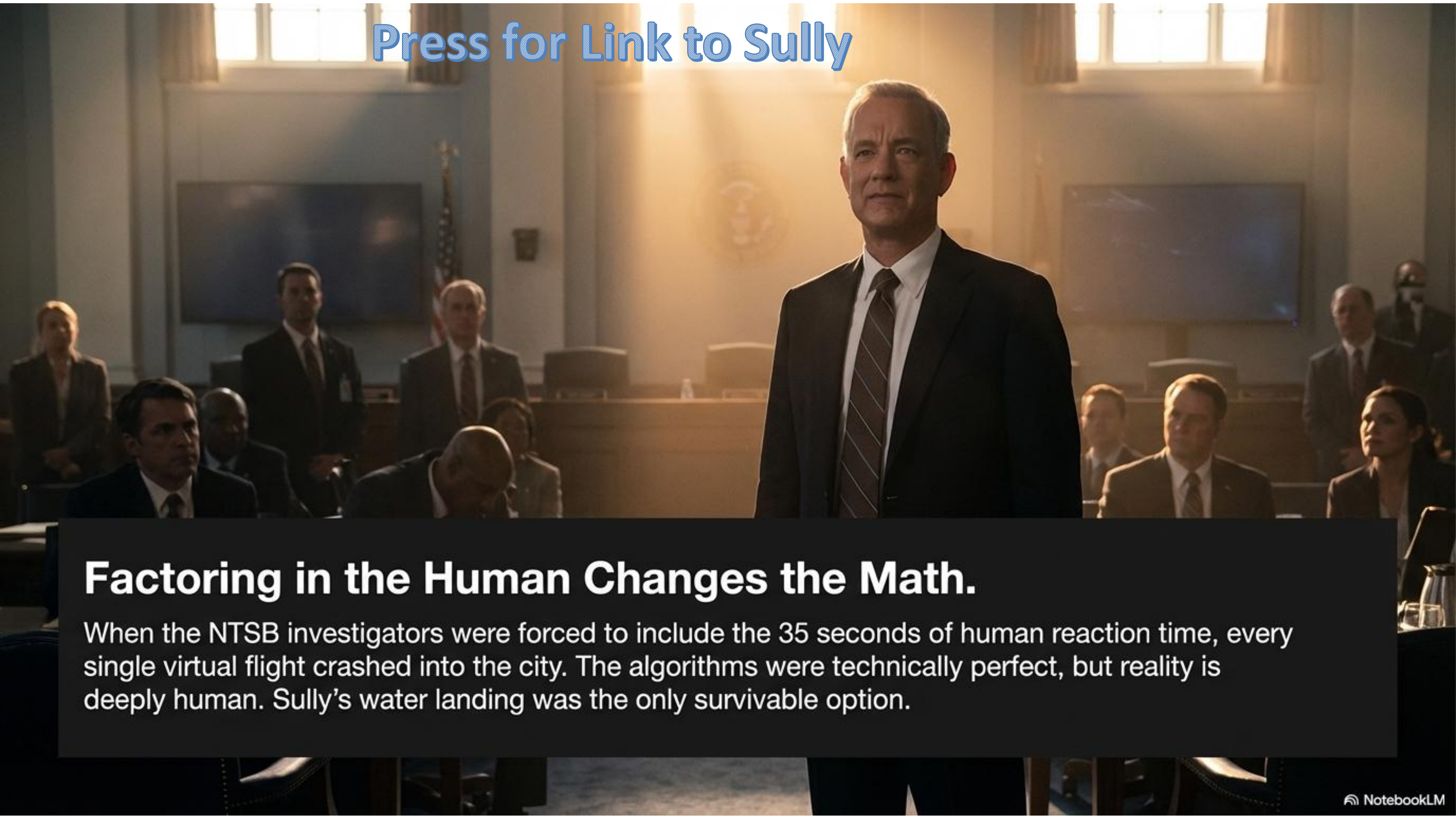
NotebookLM



The simulations show you could have made it back.

The NTSB Investigation. Algorithms and flight simulators dictate that the Hudson landing was an unnecessary risk. The computer judged the human.

NotebookLM



Press for Link to Sully

Factoring in the Human Changes the Math.

When the NTSB investigators were forced to include the 35 seconds of human reaction time, every single virtual flight crashed into the city. The algorithms were technically perfect, but reality is deeply human. Sully’s water landing was the only survivable option.

NotebookLM