

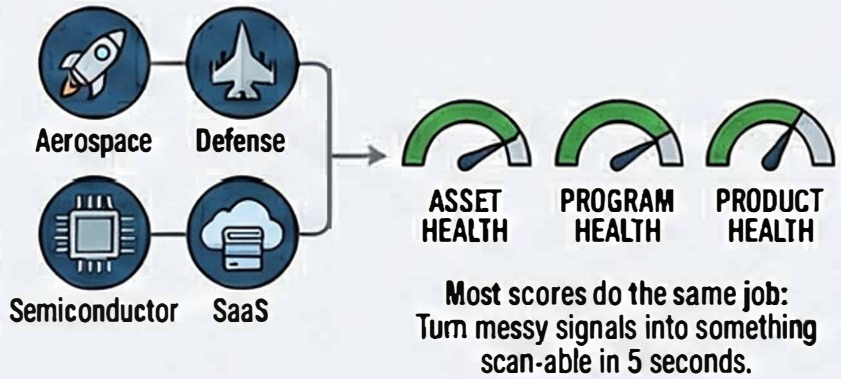
IS YOUR "HEALTH: 92%" SCORE A LIE?

UNDERSTANDING THE MATH BEHIND THE DASHBOARD ILLUSION



Looks Clean. Sounds Confident.
But if one critical dimension is falling, the single number is an illusion.

THE PROBLEM: WHY SIMPLE SCORES MASK REALITY



THE TWO CORE DECISIONS



FOUR MODELS FOR CALCULATING HEALTH (AND THEIR TRADEOFFS)

1. WEIGHTED AVERAGE – SIMPLE, FULLY COMPENSATING

$$\text{Health}_{\text{WA}} = 0.40C + 0.25F + 0.15E + 0.10V + 0.10A$$



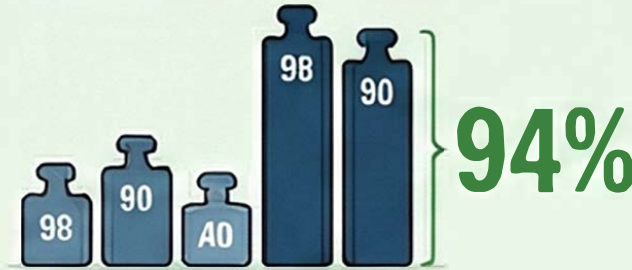
The 'School Grades' Model

Pros: Simple, auditable, easy to recreate.
Cons: Fully compensating, A 98 can quickly cancel out a 40.



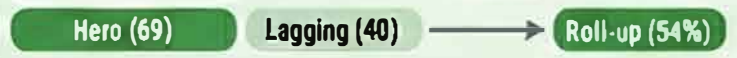
2. RMS (ROOT MEAN SQUARE) – STRONG AREAS SPEAK LOUDER

$$\text{Health}_{\text{RMS}} = \sqrt{\text{sqrt}(0.40C^2 + \dots + 0.10A^2)}$$



Highlights excellence.

Pros: Strong components impact more. Useful for highlighting standouts.
Cons: Less useful for finding weak links; rates uneven systems higher.



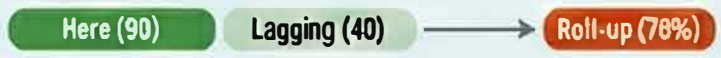
3. GEOMETRIC MEAN – NO WEAK LINKS ALLOWED

$$\text{Health}_{\text{GEO}} = C^{0.40} * F^{0.25} * \dots * A^{0.10}$$



Multiplication drops score fast with low values.

Pros: Great when every dimension must stay strong (Safety, Compliance). Cons: Harder to explain. Exponents.



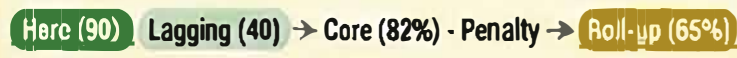
4. WEIGHTED AVERAGE + BALANCE PENALTY – PRACTICAL MIDDLE GROUND

Step 1: Core = Weighted Avg.
Step 2: Spread = Max - Min.
Step 3: Health = Core - k * Spread



Keeps simple average but penalizes unevenness.

Pros: Explainable average, refuses to let one 'hero metric' cover lagging ones. Maps to practical tool behavior.



GUARDRAILS & BEST PRACTICES: SO THE SCORE CAN'T CHEAT



FLOOR RULE: CAP HEALTH
If component < 60,
Cap Health at 75 until fixed.



ZERO RULE: FORCE LOW BAND
If must-have is 0, Force Health low regardless of other scores.



VISIBILITY RULE: SHOW SUB-SCORES
Show 4-6 components next to the roll-up.
Score = 'How we're doing'.
Components = 'What to fix'.



MASTER YOUR METRICS. LEAD WITH CONFIDENCE. OWN YOUR DATA.

Transform your dashboards into powerful strategic tools. This is how you build truly resilient systems. John C. Vines <https://www.linkedin.com/in/johncvines/>