A Personal Decision-Support System for US Equity Screening: Design, Validation, and Iteration

Project report on the design of a rule-based stock classification system, with an account of six documented errors and their corrections

Thomas · September 2026

System: public website on Cloudflare Workers · approx. 620 lines of code · all computation server-side, front end renders only

Data: TradingView Scanner API (37 fields), cross-validated against FactSet

Abstract

This report documents a US equity screening system I designed and iterated on independently during 2026. Each run scans every listed US company above USD 1.5 billion in market capitalisation, grades each one on return on invested capital and cash-flow quality, then combines that grade with price position and analyst expectations to place the company into one of nine action tiers. The output is a single layered table that can be acted on directly.

The system itself is not the main subject of this report. What the build process exposed is. I had assumed my own stock selection followed a logic, and only when I tried to write that logic as code did I find how much of it rested on intuitions I could not articulate, some of which did not survive examination. The real output of the project is a record of six errors: a floating-point boundary condition that silently dropped valid signals, an industry classification rule that failed in both directions at once, two cases of inconsistent accounting bases producing plausible but meaningless figures, a confusion between two layers of caching, and one case where my own testing method contained a blind spot. The last of these deserves separate treatment, because it shows what happens when the instrument of verification shares an assumption with the object being verified.

The report proceeds through problem definition, the methodological argument for using return on invested capital rather than price-to-earnings or margin thresholds, data source selection and validation, system architecture, the full classification ruleset, the six errors and their corrections, validation methods, the multi-timeframe problem, four shifts in how I now think about investing, and an explicit account of what the system does not do. Appendices give the complete parameter table, field list, and pseudocode for the tier cascade.

A note on how the work was done

This system was built in a continuous working pair with an AI assistant, which wrote most of the code under my direction. I set the questions, decided which sections the panel would have, which columns each table would show, and the order in which the tier rules are evaluated. Every threshold came from my own trading experience rather than from the assistant's suggestion, and the judgement calls were mine, including which com-

panies needed a manual override, how far the upper bound of the signal range should be widened, and whether the four-hour timeframe warranted a section of its own.

Corrections ran in the other direction as well, and they are reported in the body rather than left out. The Wingstop attribution error in Section 7 was the assistant's: it saw a year-over-year figure and constructed an explanation for it, I challenged that with a source reaching the opposite conclusion, and going back to the filings confirmed the assistant was wrong. It had also written stop-loss levels I have never used as though they were mine, which I rejected on the spot. Neither incident is an aside. The first produced the conclusion in Section 7 about base-period anomalies, and the second shows that a tool will present its own inference as a statement of fact about the user, who may not notice.

I state this at the front because the division of labour is part of the method rather than a footnote to it. Section 6.6 reports that verification fails when the instrument and the object share an assumption, and the same holds for a person working alongside a tool. Every threshold and rule in this report can be checked against the source code.

Contents

1. Problem Definition

1.1 A judgement process I could not reproduce

The project did not begin as a technical idea. It began with a doubt about my own reasoning. I had been investing in US equities for some time and believed I was working from a consistent standard: company quality, valuation, and chart position. When a friend asked me whether a particular stock was worth buying and I tried to explain why, what came out was a list of reasons with no stated weighting among them. Change the stock, or ask me a month later, and the structure of my reasoning changed with it.

That pointed to a specific problem. A standard that adjusts itself to fit each case is functioning as post-hoc justification rather than as a basis for the conclusion. A real standard should have one property: written down and handed to someone else, it should produce the same conclusion, and where it does not, the disagreement should be traceable to a specific condition. What I had did not behave that way.

Writing the judgement as code is the most direct test of this. Code does not accept vagueness. Writing `ROIC >= 15` forces an answer to where 15 came from. Writing "the position is right" forces a definition of right. Anything that cannot be written was never properly worked out in the first place.

1.2 Three specific failures

What pushed me to actually build the system was a set of three misjudgements I could reconstruct, each pointing at a different kind of problem.

The first was Costco. Early on I screened on net margin and, seeing Costco at around three per cent, filed it under low quality. What I did not understand at the time is that membership retail is built to earn almost nothing on the goods themselves and to make its return on membership fees and inventory turnover. The right measure there is capital efficiency, not margin. Applying a margin threshold calibrated on software to a retailer is using an instrument outside the range it was built for. This was an error of **mismatch between metric and object**.

The second was Novo Nordisk. Its price-to-earnings ratio sat visibly below comparable pharmaceutical companies, and I initially read that as a valuation advantage. Looking at analyst consensus for next fiscal year revenue shows something else: the forecast sits below current trailing twelve-month revenue. The market was not discounting the company, it was pricing stagnation. A low multiple was the conclusion rather than the opportunity. This was an error of **using historical data without forward-looking data**.

The third was Sterling Infrastructure. My chart indicator showed nothing on the weekly or daily view, while the same indicator on the four-hour chart read high. Three timeframes gave three answers for one stock, and I had been defaulting to one of them without ever deciding to. This was an error of **observing at a single scale**.

These three have something in common. None of them came from missing information. All of them came from a defect in how I was processing information, and none of the rules involved had ever been written down where I could examine them.

1.3 Scoping the project

From that analysis I set four objectives, and deliberately excluded some others that look related but are not.

Objective	What it means in practice
Rules made explicit	Every classification traces to a rule I can point at. There is no "overall impression" term.
Full market coverage	The scan covers the whole market, not only my watchlist, so the system can surface things I was not already looking at.
Traceable measurement basis	Every number has a stated source and calculation. Anomalies can be located to a specific field.
Falsifiable	Any rule can be challenged with a counterexample. Where the challenge holds, the rule changes.

Two objectives were excluded on purpose. The system does not execute trades; it outputs tiers and reasons, and the decision stays with a person. It also does not forecast returns; it does not answer how much a stock will rise, only which tier it falls into under my stated standard and why. Section 10 explains the reasoning behind both exclusions.

2. Methodology: Why Return on Invested Capital

2.1 Price-to-earnings does not measure quality

The price-to-earnings ratio divides a market price by an accounting profit. It measures how many times current earnings the market is willing to pay, which makes it a statement about pricing rather than about the business. The same ratio is consistent with two very different situations: a high-quality company temporarily mispriced, or a company whose earnings are about to decline in a way the market already anticipates. The ratio alone cannot separate them.

Novo Nordisk was the second case. In the system, price-to-earnings is therefore demoted to a reference column. It appears in the tables but takes no part in tier assignment. What does take part is return on invested capital, forward revenue growth, analyst recommendation, and the gap to the consensus target price. The ratio enters a rule at exactly one point: above 100 it triggers an avoid flag, which uses it as an outlier detector rather than as a valuation measure.

2.2 Why absolute margin thresholds fail across industries

My original quality standard was four absolute thresholds: return on equity above 15 per cent, gross margin above 40, net margin above 20, and free cash flow above 10 per cent of revenue. Each is defensible on its own and all four are common in value-investing discussion.

The trouble appears when they are applied together as hard gates. Doing so embeds an assumption: that a good business shows its quality in its margins. For software and pharmaceuticals that assumption mostly holds. For retail, distribution, and engineering contracting it produces systematically wrong answers. Costco is the clearest case, and the same problem shows up in TJX, AutoZone, and Lowe's, all of which earn their return through turnover rather than unit profit.

In methodological terms this is a **construct validity** problem. The construct I wanted to measure was business quality. The operationalisation I chose was margin. But the relationship between that indicator and the underlying construct is not stable across industries, which is to say the measure lacks cross-group measurement invariance.

2.3 ROIC and the cost of capital

Return on invested capital divides after-tax operating profit by invested capital, measuring what the business earns on each dollar committed to it. It sits closer to the construct of quality than margin does because it carries both profitability and efficiency in one number. Costco's net margin is around three per cent, but its capital turns over fast enough to produce a return on invested capital above twenty, which correctly identifies it as a good business.

The thresholds have a stated reference point. Weighted average cost of capital for large US listed companies sits broadly in the seven to ten per cent range, which means a company earning less than that on invested capital is not creating economic value through growth. Expansion at that level consumes shareholder capital rather than compounding it. The system sets eight per cent as the floor of the lowest passing grade, with the intended reading of "at least covers its cost of capital."

The 15 and 20 per cent thresholds have no comparable theoretical anchor. They were set after looking at a set of companies I knew well, and correspond to "clearly above cost of capital" and "shows some structural advantage." That is a judgement call, and Section 10 lists it among the system's limitations.

2.4 Four supplementary thresholds and the dual-path design

The original four margin thresholds were not discarded. They were converted into an alternative path. The final grading rule is:

Grade	Path one (capital efficiency)	Path two (four thresholds)
A	ROIC ≥ 20%	all four satisfied
B	ROIC ≥ 15%	three of four
C	ROIC ≥ 8%	two of four
D	none of the above	

The two paths are disjunctive; satisfying either one is enough. The reasoning is that the capital-efficiency path handles low-margin, high-turnover businesses correctly, while the threshold path catches companies whose return on invested capital is understated for accounting reasons but whose earnings quality is genuinely high. Each path compensates for a failure mode of the other.

The four thresholds themselves are unchanged. What changed is that they now function as a count rather than as simultaneous hard gates, and that change was the first substantive methodological revision in the project.

2.5 From historical to forward-looking measures

What the Novo Nordisk case exposed is that every trailing twelve-month measure describes the past. Changes in a business appear first in analyst expectations and only later in reported results. A system reading only reported results responds to an inflection at least one quarter late.

I therefore added two forward fields: consensus revenue and consensus earnings per share for the next fiscal year. From these the system derives forward revenue growth and a forward price-to-earnings ratio. Forward revenue growth is written directly into the tier cascade: below negative four per cent, a company drops into the risk tier regardless of its quality grade, with an explanatory note attached.

The threshold is not set at zero, because fiscal-year consensus carries noise of its own. Companies have different fiscal year ends, and consensus figures move as analyst coverage changes. The noise band I observed runs to roughly plus or minus two per cent. Setting the threshold at negative four clears that band while still producing a text warning, without a tier change, anywhere between negative four and zero. This trades false positives against false negatives in the direction of not penalising a company on noise alone.

3. Data Engineering

3.1 Source selection

The system needs three categories of data: fundamentals (returns, margins, growth, balance sheet), price and technical position (moving averages, relative strength, period highs and lows), and market expectations (target prices, recommendations, next-fiscal-year consensus). The third category is the constraint, because it cannot be derived from public filings and has to come from a vendor that aggregates broker estimates.

I looked at several options. Professional terminals have the best data and an unrealistic cost. Free financial sites generally lack consensus estimates or have incomplete field coverage. I settled on TradingView's Scanner endpoint because it returns all three categories in a single request, covers enough fields, and has a stable response format.

The call is a POST to `scanner.tradingview.com/america/scan` with a body specifying filters, requested columns, sort order, and a pagination range. One implementation detail is worth recording: setting the request `Content-Type` to `text/plain` rather than `application/json` causes the request to be treated as a simple request, avoiding a cross-origin preflight. It looks like a trivial change, but it is one of the reasons the system runs at all in a serverless environment.

3.2 Validating analyst consensus availability

The first thing I did after selecting the source was test whether the forward fields could be trusted. This turned out to be the single most valuable piece of work in the project, because until then forward estimates had to be looked up by hand, which was the system's largest gap.

The test took a set of companies I had already researched on FactSet and compared TradingView's `earnings_per_share_forecast_next_fy` and `revenue_forecast_next_fy` against the FactSet consensus, one by one. For AppLovin, TradingView returned next-fiscal-year EPS of 15.957 against a FactSet consensus of 15.95, a difference under one tenth of one per cent. Deviations across the remaining sample were similarly small.

What this establishes is that TradingView's forward fields draw on the same broker estimate pool as FactSet, or one that overlaps heavily with it. The validation converted a per-stock manual lookup into an automatic market-wide scan, and made it possible for the system to flag the Novo Nordisk pattern of a low multiple concealing expected stagnation.

> There is a methodological point here. Validating a data source costs far less than what it prevents. Roughly two hours of comparison replaced a manual lookup that would otherwise repeat for every stock I researched. More to the point, using the fields without validating them would have produced numbers that look precise and come from an unknown place, which is a worse position than having no number at all.

3.3 Scan universe and field list

Five filters define the universe: exchange restricted to NYSE, NASDAQ, and AMEX; primary listing only; instrument type equal to stock; type specification containing "common"; and market capitalisation above USD 1.5 billion.

The fourth filter was added later. Without it, preferred shares entered the scan. Preferred share tickers look much like common stock tickers, but the financial metrics do not apply to them, so fields such as price-to-earn-

ings come back empty or meaningless. Contamination of this kind is hard to see in the output because it does not raise an error. It quietly distorts the distribution.

The USD 1.5 billion floor is a liquidity choice. Below that, analyst coverage thins out and consensus estimates often rest on one or two brokers, which degrades the reliability of the forward fields. The system separately flags any stock whose ten-day average dollar volume falls below USD 30 million as thinly traded.

The request asks for 37 fields in five groups. Pagination is 400 rows per call, with the loop capped at 2,800 and an early exit when a page returns fewer than 400.

Group	Fields
Identity and scale	name, market_cap_basic, industry, total_revenue_ttm, average_volume_10d_calc
Earnings quality	return_on_invested_capital, return_on_equity, gross_margin_ttm, net_margin_ttm, free_cash_flow_margin_ttm, total_revenue_yoy_growth_ttm, price_earnings_ttm
Balance sheet	cash_n_short_term_invest_fq, total_debt_fq, free_cash_flow_fq, free_cash_flow_ttm, total_current_assets_fq, total_current_liabilities_fq
Price and position	close, SMA50, SMA200, EMA100, RSI, High.6M, price_52_week_low, price_52_week_high, Low.3M, Low.1M, Perf.1M, Perf.Y, Perf.3Y
Expectations and calendar	price_target_1y, recommendation_mark, revenue_forecast_next_fy, earnings_per_share_forecast_next_fy, earnings_release_date, earnings_release_next_date

The system also carries a watchlist of just over five hundred tickers, grouped by sector and by how closely I follow them. The list does not filter anything. It only annotates results, marking which companies I was already tracking and which the scan surfaced that I had never looked at. The latter are tagged "new." The point of that design is to give the system a chance to overturn my attention, rather than confirming what I already believed within a universe I had already chosen.

3.4 Data freshness and two-layer caching

Price data arrives either in real time or on a fifteen-minute delay depending on the exchange. Fundamental metrics update with reported results, so quarterly. Target prices and recommendations update daily. Forward consensus moves as analysts revise, also typically daily.

The page displays current market state, distinguishing weekend, pre-market, open, and closed, and states explicitly which point in time the displayed prices reflect. I added this after noticing that I would look at the panel on a Saturday and read the numbers as current. Having the system state the temporal status of its own data is more reliable than expecting the reader to infer it.

Caching operates in two layers, and Section 6.5 describes what once went wrong with that arrangement. The edge cache holds results for fifteen minutes to avoid repeating a full market scan. The response sent to the browser is explicitly marked as non-storable. A force-recompute entry point exists but is rate-limited to once

every ninety seconds, since the site is public and no visitor should be able to trigger unlimited full-market scans.

4. System Architecture

4.1 Why serverless

Three considerations drove the choice. There is no server to maintain, which keeps the project from turning into an ongoing operational commitment. The deployment path is very short: edit the code, paste it into the console, deploy, and every visitor sees the new version immediately. And it resolves cross-origin issues by construction, because the call to TradingView originates server-side.

A fourth benefit only became clear later. A serverless environment forces state to a minimum, and the entire system holds exactly one global variable, used for rate-limiting forced recomputation. Less state means fewer places for things to go wrong.

4.2 Separating computation from presentation

Every formula, threshold, tier rule, and manual override executes server-side. What reaches the browser is a finished result: a grade, a tier, and a line of explanatory text per stock. Front-end code handles rendering, sorting, filtering, and section folding, and contains no decision logic.

The primary reason is that the site is public and I would rather the parameter table not be readable from it. An unanticipated benefit followed. Because the front end has no access to the formulas, I cannot debug by tweaking a number in the browser to see what happens. Any parameter change has to go back to the server code and be redeployed, which turns every adjustment into a recorded change.

4.3 A single source of truth for column definitions

The panel has four tables with different fields, and I add and remove columns often. Originally the header row and the data row were written as two separate blocks of code. Changing one and forgetting the other produces a misalignment where a column of numbers sits under the wrong heading. Nothing raises an error, and the output looks entirely normal.

I replaced this with a central column-definition object in which each column registers its display name, sort key, alignment, and value function. Each of the four tables holds only an array of field names, and both header and data rows are generated from the same definition.

```
const CD = {
  n:    { th:'Ticker',  L:1, cell: r => ... },
  Qn:   { th:'Quality', sk:'Q', cell: r => ... },
  roic: { th:'ROIC',    cell: r => ... },
  ...
};
const T2 = ['n','tier','Qn','roic','rg','fg','pe','fpe', ...];   // tiered action table
const T4 = ['n','g','tier','Q','roic','rg','fg','pe', ...];      // full market table

function buildHead(id, keys) { ... generated from CD ... }
function buildRow(r, keys)   { ... generated from CD ... }
```

Adding or removing a column is now a one-element edit to an array, and header and data cannot disagree. This is not a performance change. It moves an entire class of error from possible to structurally impossible, and the return on it arrives not at the time of the change but in every subsequent modification that no longer carries that risk.

5. The Classification System

Classification runs in three layers. Quality answers whether the business is good. Position answers whether now is the time. Signals answer whether the stock has reached some extreme. The three combine into a tier. Quality changes quarterly; position and signals change daily.

5.1 Quality grading

The rule appears in Section 2.4. Two details need adding.

A company with no financial data at all, typically a recent listing or one without analyst coverage, receives a D grade, but the system records a separate flag and writes the explanation as "financial data unavailable, not a quality judgement." Without that distinction, new listings sit alongside genuinely weak companies, and the two require quite different handling.

Financials are isolated entirely. Banks, insurers, real estate investment trusts, and asset managers run balance-sheet businesses for which return on invested capital carries no meaning. These go into their own tier, with text stating that they should be assessed on net interest margin, non-performing loan ratios, funds from operations, and similar measures. The system does not pretend to evaluate them.

5.2 Position

Position uses three quantities: distance from the 200-day moving average, relative strength index, and one-month price change. Two exclusion states come first:

- Overheated: RSI at or above 66, or a one-month gain at or above 20 per cent
- Oversold: RSI at or below 35, or a one-month decline at or above 15 per cent

Three positions follow:

- **Buy zone**: above the 200-day average, allowing up to 2 per cent below it, with RSI at or above 38, and neither overheated nor oversold
- **Wait zone**: does not meet the buy zone test, but sits no more than 15 per cent below the 200-day average, and neither overheated nor oversold
- **Observe**: everything else

Overheating is excluded separately because a company can be high quality and above its moving average while having just risen thirty per cent in a month, at which point the risk-reward of initiating a position has clearly deteriorated. The rule exists to stop the system from recommending accumulation into a sharp run-up.

5.3 Signals

Signals approximate a volume and price extreme indicator I use on charts. The system reproduces its logic through three conditions measured against the 52-week low, with a shared precondition that room remains above, meaning the six-month high sits at least 25 per cent higher than the current price.

Signal	Condition	Reading
Blue bar	within 3% of the 52-week low	at an extreme low, including breaks below
Recent exit	low set within the past three months, now 3% to 18% above it	just leaving the base
Approaching	3% to 6% above a low set some months ago	early warning, weaker than the other two

The upper bound on the recent exit condition was originally 12 per cent and was widened to 18. The evidence was three counterexamples: TSLA at 12.2, GPOR at 13.5, and LULU at 14.2, all of which showed the signal on the chart but were cut off by the 12 per cent ceiling. After widening, I rechecked the sample that had already been inside the range and found no obvious new false positives.

A separate freshness test asks whether the 52-week low was set within the past three months. That window was originally one month, which missed cases such as LULU, where the low had been set about two months earlier.

5.4 The tier cascade

Nine tiers are evaluated in strict order and the first match stops evaluation. The order carries information in itself: the earlier a condition sits, the more it functions as a veto.

1. Manual override → the tier specified in the note (events no field captures: price competition against a core product, guidance cuts, pending litigation)

2. Financial → separate tier (banks, insurers, REITs, asset managers; ROIC does not apply)

3. Cash-burning with cash at risk → avoid (runway under 6 quarters, or net debt with a current ratio below 1)

4. Cash-burning but well funded → early-stage tier (ratios distorted; read runway and market cap less net cash instead)

5. Avoid → grade D, or negative ROIC, or negative growth combined with cash burn, or P/E above 100

6. Risk → forward revenue growth ≤ −4%, or recommendation weaker than 1.9, or upside below 5%

7. **Tier 1, accumulate in tranches** → grade A or B, buy zone, upside ≥ 12%, recommendation ≤ 1.65

8. **Tier 2, await signal** → grade A or B, and either a blue-bar or recent-exit signal, or the wait zone with upside ≥ 20%

9. **Tier 3, observe** → everything else

Figure 1. The tier cascade. Rules are evaluated top to bottom and the first match ends evaluation.

Two points need explanation.

The risk test at position 6 sits ahead of tiers 1 and 2 because a forward revenue decline is a veto condition, and a high quality grade should not override it. A company with excellent returns on capital whose next-year revenue is expected to fall is one whose returns are becoming historical. Rating it highly on trailing metrics at that moment is wrong. Placing the veto first encodes the rule that quality does not absolve expectations.

Manual override sits first because it handles precisely the information no field reaches. Ten override records currently exist, each with a stated reason, covering cases such as a core product facing low-price competition or a high return on equity that turns out to be a buyback effect on the denominator. None of this can be derived from any field. Making the override an explicit layer rather than quietly adjusting the rules keeps it auditable: at any time I can list exactly which ten judgements are mine personally and why each one is there.

5.5 A counterintuitive result

Once the classification was working, a result appeared that I had not anticipated: **an A grade does not imply tier 1**.

Quality and tier are separate dimensions. Quality answers whether a company is worth owning. Tier answers whether now is the moment. An A-grade company sitting at its 52-week low lands in tier 2 rather than tier 1, because the decline has not been confirmed as over, and buying into a base is a different approach from buying after trend has resumed. An A-grade company that has climbed back into the buy zone with adequate upside and a strong recommendation is what reaches tier 1.

The result matters because it corrects a habit of mine. I had tended to buy once I liked a company's quality, collapsing "good company" and "buy now" into one judgement. Separating the dimensions lets me ask a sharper question: am I paying for quality here, or taking a risk on position.

6. Iteration Record: Six Errors and Their Corrections

This is the section I consider most valuable. The six errors fall into different categories and required different kinds of correction, so I have ordered them by type rather than chronologically.

6.1 Boundary error: floating-point arithmetic silently dropping signals

Symptom. Some stocks I had confirmed on the chart as showing a blue-bar signal were not flagged by the system. Not many, and with no obvious pattern.

Cause. The blue-bar condition is "within 3 per cent of the 52-week low," written as `d52 <= 3` where `d52 = (px / l52 - 1) * 100`. When a price sits exactly three per cent above the low, floating-point arithmetic does not return a clean 3. It returns 3 followed by a run of zeros and then a very small non-zero digit. That value exceeds 3, the test evaluates false, and the signal disappears.

Why it was hard to find. Nothing errors. No outlier appears. One record is quietly missing at a boundary. Had I not been comparing stock by stock against charts, the problem could have persisted indefinitely.

Correction. Tolerance-based comparison functions, applied uniformly to every threshold comparison in the system:

```
const le = (a, b) => a <= b + 1e-9;
const gt = (a, b) => a >  b + 1e-9;
const ge = (a, b) => a >= b - 1e-9;
```

I then built thirteen test cases landing exactly on thresholds and confirmed all pass after the fix.

What this error shows. Screening systems fail in two shapes. They can output something they should not, which is easy to catch because the result looks wrong when you read it. Or they can fail to output something they should, which is nearly impossible to catch by reading results, because absent items do not announce themselves. System design should treat the second case with more suspicion, and the only way to test for it is to check against known answers.

6.2 Classification error: an industry rule failing in both directions

Symptom. During a routine check I found that Morgan Stanley, BlackRock, and Apollo Global Management were not being routed into the financial tier, so their returns on invested capital were feeding into ordinary tier assignment. At the same time CME, ICE, and Nasdaq had been incorrectly classified as financials.

Cause. The original isolation rule ran a regular expression against the industry label. Testing revealed the actual label values differ from what I had assumed: Morgan Stanley, BlackRock, and Apollo carry Investment Managers; Ares Capital carries Financial Conglomerates; Blackstone Secured Lending carries Investment

Trusts. None of these were covered. Meanwhile the pattern included the word "Banks," which appears in the industry description of exchange operators, catching them by accident.

Correction. A three-layer structure:

1. A broad regular expression covering six label families: real estate investment trusts, banks, insurance, investment managers, financial conglomerates, investment trusts
2. A whitelist exempting eighteen companies caught by the pattern that are in fact fee-based operators. Exchanges, payment networks, pure fee-earning asset managers, and insurance brokers earn transaction fees rather than running a balance sheet, so ROIC does apply to them
3. A blacklist adding five companies whose industry label does not catch them but whose business is balance-sheet lending, mainly consumer credit

What this error shows. The mistake was not a coding one. I had treated my own mental model of industry classification as though it were the data's actual values. When a rule depends on an external enumeration, the values in that enumeration have to be checked empirically rather than inferred from common sense. The corrected three-layer structure is not elegant, but it concedes something true: the industry labels are defined by someone else, and my rule has to accommodate them rather than the reverse.

6.3 Measurement basis error I: inconsistent revenue bases

Symptom. StoneCo showed forward revenue growth of positive 404 per cent.

Cause. Forward growth divides consensus next-fiscal-year revenue by current trailing twelve-month revenue and subtracts one. For most companies that is fine. But payment, credit, and insurance companies sometimes report revenue on a net basis, after deducting amounts paid to third parties, while analyst forecasts are issued on a gross basis. Dividing one by the other produces a number with no meaning.

What makes this dangerous is that the output looks like a precise measurement. An empty field tells the reader there is no data. Positive 404 per cent reads as a finding.

Correction. A basis-anomaly test. Either condition flags the case, blanks the forward growth figure, and displays a basis-mismatch marker:

- Absolute net margin above 100 per cent, since net income exceeding revenue is not possible on a consistent basis
- Forecast-to-actual ratio above 2.5 or below 0.4

With the test in place roughly one hundred stocks are flagged market-wide. I checked eight known cases and all eight were identified correctly.

What this error shows. Two numbers should be confirmed to share a basis before being divided. That is elementary in financial analysis, but when a calculation is written into code and applied to several thousand com-

panies, the elementary check does not happen by itself. The system has to test for basis explicitly, because it has none of the human sense that a number looks wrong.

6.4 Measurement basis error II: scale disguised as early stage

Symptom. Air Products, a long-established industrial gases company, was classified as early-stage cash-burning.

Cause. The original test for the early-stage tier was negative free cash flow together with a negative net margin. Air Products took a writedown in one period that pushed both negative, and the test fired. The same thing happened with ECHO, a company carrying USD 16 billion in net debt, which does not fit any sense of "early."

Correction. A scale gate requiring revenue below USD 800 million or a net margin below negative 30 per cent. The first clause excludes occasional losses at mature companies; the second retains companies that have grown past that revenue level but whose loss magnitude is genuinely characteristic of an early-stage business.

A related problem surfaced at the same time. Runway is cash divided by quarterly burn, and a quarter with only a small loss produces a runway of four hundred quarters or more. MicroStrategy did exactly that. Four hundred quarters carries no information and only creates confusion, so runway is now capped at forty.

What this error shows. Where a category's name ("early stage") and its test (two values negative) leave a semantic gap, some edge case will eventually open that gap. The fix is not to adjust a threshold but to supply the omitted part of the meaning, which here was the requirement that the company still be small.

6.5 Infrastructure error: confusing two layers of cache

Symptom. After adding a pullback pool for leading stocks, the page showed zero matches. I had already confirmed the rule itself was correct.

Diagnosis. This took three rounds. First I assumed the edge cache held an older version of the data, so I added a code fingerprint to the cache key, ensuring any deployment invalidates the old cache automatically. Still zero. Second, I called the endpoint directly and confirmed the pullback flags were present and correct, showing 45 and 30 matches. The data was right and the page was wrong. The third round found the actual cause.

Cause. The response carried a cache header with the fifteen-minute lifetime intended for the edge cache. The browser read the same header and stored the JSON locally for fifteen minutes. After a deployment the page code was new while the data came from the browser's cache, and the older payload had no pullback fields, so the filter returned nothing.

Correction. Separate cache headers for the two layers. The copy written to the edge cache carries a lifetime; the copy sent to the browser is explicitly non-storable. The front end also appends a timestamp parameter and declares no-store on its own request.

```
// stored to the edge cache: carries max-age
ctx.waitUntil(cache.put(key, new Response(body, {
  headers: { "Cache-Control": "public, max-age=" + P.cacheMinutes * 60 }
})));

// sent to the browser: must be no-store
return new Response(body, {
  headers: { "Cache-Control": "no-store, must-revalidate" }
});
```

6.6 Observer error: a blind spot in the testing method

The reason the previous problem took three rounds to diagnose deserves its own section.

When testing the endpoint I habitually appended a timestamp parameter to guarantee fresh data. The habit is sound in itself, but it meant every one of my test requests bypassed the browser cache. Put differently, **the method I was using to verify the system happened to be immune to the exact fault the system was committing**. Every test showed a correct result, while an ordinary page load, including my own, showed a wrong one.

Of the six errors this is the one I consider most valuable, because it concerns neither finance nor code but verification itself. When the instrument of verification shares an assumption with the object being verified, verification fails at exactly the point where that assumption breaks. Management research has a counterpart in common method bias, where measuring independent and dependent variables through a single instrument produces a correlation that belongs to the method rather than the phenomenon. The form differs and the structure is the same: the measurement is not independent of what it measures.

The practical rule I took from this is that verification should include at least one channel identical to the ordinary usage path, rather than running entirely through cleaner test channels. A test environment that is cleaner than production is itself a source of bias.

7. Validation Methods

Validation runs in four forms, each catching a different class of problem.

Method	Procedure	What it catches
Boundary testing	Inputs constructed to land exactly on each threshold; thirteen cases	Floating-point error, inverted comparison operators, open and closed interval mistakes
Counterexample calibration	Collect stocks I confirmed on charts as showing a signal that the system did not flag, then trace each	False negatives, the signals that were dropped
Cross-validation	Compare system values against the corresponding FactSet values one by one	Source basis problems, misunderstood field definitions
Structural self-check	Verify every key in each table's field array exists in the column definition; verify every render function guards against unloaded data	Inconsistencies introduced by refactoring, null reference errors

Counterexample calibration has been the most productive. The procedure is to judge a set of stocks independently on charts, then compare against system output and attend only to the disagreements. Widening the signal range from 12 to 18 per cent came out of exactly this. The most recent run compared fourteen samples with no disagreements.

One correction is worth recording, running from me to the assistant (see the note on how the work was done, above). While analysing Wingstop the assistant saw a year-over-year earnings per share figure of negative 29.6 per cent, read it as evidence of declining profitability, and proposed several possible causes. I challenged that with a source reaching the opposite conclusion, that profit was still growing. Going back to the filings confirmed the assistant was wrong: net income and earnings per share were both up, and the negative 29.6 figure came from a one-off gain in the prior-year base.

The lesson is more specific than "check the data." The error was to see a year-over-year figure and immediately explain it, without first asking whether the base period contained anything unusual. A year-over-year change is a difference between two points, and the anomaly can sit at either end. Faced with a surprising year-over-year number, the first move should be to decompose both ends rather than to construct an explanation. Constructing an explanation is itself what leads to premature belief that the number is real.

8. The Multi-Timeframe Problem

The Sterling Infrastructure case ran as follows. One indicator, three readings: 0.88 on the weekly chart, 5.43 on the daily, 29.03 on the four-hour. The spread exceeds an order of magnitude.

The indicator was not malfunctioning. Its lookback is a fixed number of bars, so the actual time span it covers differs entirely by timeframe. One hundred and fifty weekly bars is roughly three years, the same count on a daily chart is about seven months, and on a four-hour chart about three to four months. A stock can be far from

its low on a three-year scale and at its low on a three-month scale at the same time without contradiction. The readings describe relative position over different windows.

The problem was that the system used only a 52-week basis, so four-hour-scale opportunities were invisible to it. Sterling sits 69 per cent above its 52-week low, which means daily and weekly signals will never fire.

The obvious fix is to add "within 3 per cent of the three-month low" as a condition, but this returns about one hundred stocks, mixing ordinary pullbacks in leading stocks with names in sustained decline. The two are indistinguishable on that single criterion while implying opposite actions.

Two further conditions separate them: a year-to-date gain of at least 20 per cent, and a position no worse than 25 per cent below the 52-week high. The first requires the stock to have actually led this year, the second requires the pullback to have real depth. Together they narrow one hundred to forty-five, and reviewing them individually, all are deep pullbacks in leading stocks, Sterling among them.

The pullback pool occupies its own section in the panel rather than mixing with bottom signals, because the two carry different operating logic. A bottom signal is a bet on trend reversal; a pullback is continuation of an existing trend. The first invalidates on a break of the 52-week low, the second on a break of the three-month low. Placing logically different things in one table leads a reader to treat them the same way without deciding to.

There is a methodological point in this. I first framed it as a technical question of whether to add a four-hour timeframe, with a yes or no answer. It was actually a translation problem. What the four-hour chart supplies is relative position over a shorter window, and that information can be expressed using three-month highs and lows already present in daily data, without retrieving four-hour bars at all. Understanding what an indicator measures turned out to matter more than understanding what interval it runs on.

9. Four Shifts in How I Think About Investing

9.1 From finding bargains to assigning tiers

My original approach was to look for things the market had underpriced, using low valuation multiples as the test. After building the system I work differently: establish which tier a company falls into, then decide how that tier should be handled.

The difference is that the first approach assumes an objective fair value exists and that my job is to detect deviations from it. The second makes no such assumption. It only recognises that companies in different states call for different responses: tier 1 can be accumulated in tranches, tier 2 waits for a signal, tier 3 is only observed, the risk tier needs individual judgement, and the early-stage tier is read on cash runway rather than on any ratio.

This lowered the system's ambition, which I think was correct. It no longer attempts to answer what a stock is worth. It answers what state the stock is in under my stated standard. I can answer the second question and I cannot reliably answer the first. Narrowing the question to what I am able to answer was one of the more useful adjustments the project produced.

9.2 From absolute metrics to capital efficiency

The Costco misjudgement changed more than which metric I use. It made clear that every metric carries an implicit assumption about what a good business looks like, and that the assumption has a range of validity. Margin assumes a good business earns on unit profit, which holds for software and fails for retail.

Moving to return on invested capital did not find an assumption-free metric. It has assumptions of its own, and treats asset-heavy and asset-light businesses somewhat asymmetrically, while distorting for companies carrying large goodwill balances. The difference is only that its range is wider and covers more industries. Keeping the four margin thresholds as a second path is a concession to the same point, that no single metric covers every case.

9.3 From judgement to rules

This is the most fundamental of the four. Writing judgement as rules gives up flexibility and returns two things.

The first is traceability. When the system assigns tier 2, I can immediately identify the rule responsible. If I disagree with the conclusion, the disagreement has been located to that rule instead of remaining at the level of a feeling that something is off.

The second is accumulation. Unwritten judgement does not accumulate, because it re-forms each time and the lesson from the previous occasion does not carry forward on its own. Written as rules, each correction stays in the system and the same mistake does not recur. The record of six errors is itself the vehicle for that accumulation.

The cost is real. Rules produce mechanical answers on edge cases, and the ten manual overrides are the part the rules cannot handle. But making those exceptions an explicit layer that can be enumerated and audited is better than letting them hide inside an overall judgement. At minimum I know how much of my own opinion is in the system and what the stated reason for each piece is.

9.4 From conviction to falsifiability

At the start I wanted a system that produced correct answers. I now think its value lies in the fact that every answer it produces can be argued against specifically.

Concretely: if the system says tier 1 and I disagree, the disagreement necessarily sits in the quality grade, the position test, the upside, or the recommendation. I can name which one, argue for how that rule should change,

and then watch the classification of several thousand stocks shift together so that I can see immediately whether the change introduced a new problem.

That property makes the system something that improves over time rather than a device for confirming what I already thought. Tagging non-watchlist stocks as "new," described in Section 3.3, follows the same reasoning: a system operating only within the universe I already attend to can never tell me what I have been missing.

10. Limitations

This section states what the system does not do. I regard it as no less important than what precedes it, since a tool that does not declare its boundaries gets applied where it does not hold.

No genuine backtest is possible

The data source provides current values rather than point-in-time values. I cannot retrieve what a stock's return on invested capital looked like as of a date last March; I can only retrieve today's figure. Without point-in-time data a rigorous backtest is not possible, since evaluating past decisions with present data introduces look-ahead bias. All validation in the system therefore concerns rule consistency, not returns. The system has not demonstrated that it makes money and does not claim to.

The watchlist carries selection bias

The five hundred or so tickers come from what I was already following, which is itself the output of an earlier screening performed with a standard that had not yet been worked out. Scanning the whole market mitigates this, though the list still shapes where my attention goes when reading results. The "new" tag is a partial remedy with limited effect.

Thresholds are set by judgement

The 15 and 20 per cent quality thresholds, the 12 per cent upside floor, the 1.65 recommendation ceiling, and the 20 per cent monthly gain limit were calibrated from experience rather than optimised. I have not run a sensitivity analysis and do not know how much the output would shift if the tier 1 upside floor moved from 12 to 15. This remains open.

Consensus estimates carry their own bias

Analyst forecasts are well known to skew optimistic, particularly for smaller companies with thin coverage. Taking forward growth as an input inherits that skew. The current treatment is only the USD 1.5 billion market capitalisation floor, which screens out the most thinly covered names. That mitigates rather than resolves the issue.

What the system does not address at all

It makes no assessment of business model durability, management quality, competitive dynamics, or regulatory and policy exposure. All ten manual overrides fall into exactly this category, which suggests information of this kind matters a good deal and simply lies outside what the system can reach. Treating its output as a complete judgement would be a misuse. It handles the quantifiable portion so that attention can go to what remains.

11. Connections to Management and Organisational Decision-Making

The subject matter here is investing, but several problems encountered in the process have counterparts in management research and organisational practice. I list them as connections I noticed rather than as arguments by analogy.

Codifying tacit knowledge, and what it costs. Turning my stock selection judgement into rules is a case of converting individual tacit knowledge into explicit knowledge that an organisation could use. The gains are definite: transferability, auditability, accumulation. So are the losses, chiefly a reduced ability to handle edge cases, which the ten manual overrides exist to compensate for. This is the same question the knowledge management literature raises about codification, and my experience is that both the gain and the loss are more concrete than I expected.

Construct validity and measurement invariance. Margin failing as a measure of quality across industries is, methodologically, a failure of cross-group measurement equivalence. All three of my measurement-basis problems belong to one family: margin across industries, net versus gross revenue, and the semantic gap in the early-stage definition. In each, the correspondence between the operational indicator and the intended construct broke down within some subgroup. Checking measurement equivalence before making cross-group comparisons is a habit this project established for me.

The general form of common method bias. The testing blind spot in Section 6.6 is structurally the same as common method bias: the instrument and the object shared a source, which made the bias invisible. My practice now is to deliberately retain one verification channel identical to real use, even where it is less clean.

Locating disagreement. This is the point I think generalises furthest. The real value of making rules explicit is not that it guarantees correct conclusions but that it makes disagreement locatable. When two people differ on a stock and both are working from the same explicit rules, the difference necessarily rests on one identifiable rule, and the discussion has somewhere to go. When both are working from overall judgement, the discussion can only exchange positions. That property holds in any setting requiring a decision among several people, and it holds independently of whether the conclusion turns out to be right.

Appendix A. Parameters

Parameter	Value	Meaning and basis
mcapMin	USD 1.5bn	Scan floor. Below this, coverage thins and consensus reliability falls
blueLowPct	3%	Blue bar: maximum distance above the 52-week low
blueRoomPct	25%	Shared precondition for all three signals: minimum room to the six-month high
rExitLo / rExitHi	3% / 18%	Recent-exit rebound band. Upper bound was 12%, widened on three counterexamples (TSLA, GPOR, LULU)
nearLo / nearHi	3% / 6%	Approaching zone, warning only
fgRisk	−4%	Forward revenue growth demotion threshold. Not zero, to clear roughly ±2% fiscal-year noise
cacheMinutes	15	Edge cache lifetime
Force-recompute limit	90 s	Site is public; prevents unlimited visitor-triggered full scans
Quality A / B / C	20 / 15 / 8%	ROIC. The 8% floor references the lower end of large-cap US WACC
Tier 1: target upside / rec	≥12% / ≤1.65	Upside is the gap from current price to the one-year consensus target. Recommendation is a numeric scale on which lower values indicate stronger buy. Both calibrated from experience, with no sensitivity analysis performed
Tier 2: wait-zone upside	≥20%	Higher bar compensates for the weaker position
Overheated / oversold	RSI 66 / 35	With one-month gain ≥20% or decline ≥15%
Buy zone	SMA200 −2%	Slightly below the average is allowed; RSI must be ≥38
Cash at risk	runway <6 qtrs	Or net debt with current ratio <1
Runway cap	40 qtrs	A marginal loss quarter yields 400+; no information content, so capped
Early-stage scale gate	revenue <USD 800m	Or net margin <−30%. Prevents mature companies with one-off losses being misclassified
Basis anomaly test	ratio >2.5 or <0.4	Or absolute net margin >100%
Thin liquidity flag	USD 30m	Ten-day average dollar volume below this
Floating-point tolerance	1e−9	Applied to every threshold comparison

Appendix B. Field List (37 items)

See the grouped table in Section 3.3. All fields are retrieved in a single request, paginated at 400 rows per call with a loop cap of 2,800.

Appendix C. Tier Cascade Pseudocode

```
// quality grade
q4 = 0
if (ROE > 15)              q4++
if (gross margin > 40)     q4++
if (net margin > 20)       q4++
if (FCF / revenue > 10)    q4++

if      (ROIC >= 20 || q4 == 4)  Q = "A"
else if (ROIC >= 15 || q4 == 3)  Q = "B"
else if (ROIC >=  8 || q4 == 2)  Q = "C"
else                             Q = "D"

// position
hot  = RSI >= 66 || monthly gain >= 20
cold = RSI <= 35 || monthly loss >= 15
if      (vs SMA200 >= -2  && !hot && !cold && RSI >= 38)  pos = "buy"
else if (vs SMA200 >= -15 && !hot && !cold)               pos = "wait"
else                                                       pos = "observe"

// tier cascade (first match stops)
if      (manual override exists)                      t = override tier
else if (financial)                                   t = "financial"
else if (burning cash && cash at risk)                t = "avoid"
else if (burning cash)                                t = "early stage"
else if (Q == "D" || ROIC < 0
         || (growth < 0 && FCF < 0) || PE > 100)     t = "avoid"
else if (rec > 1.9 || upside < 5 || fwd growth <= -4) t = "risk"
else if ((Q == "A" || Q == "B") && pos == "buy"
         && upside >= 12 && rec <= 1.65)              t = "tier 1"
else if ((Q == "A" || Q == "B")
         && (blue bar || recent exit
             || (pos == "wait" && upside >= 20)))     t = "tier 2"
else                                                  t = "tier 3"
```

Appendix D. Error Summary

Type	Symptom	Cause	Correction
Boundary	Some blue-bar signals not flagged	Floating-point arithmetic makes a value exactly at the threshold slightly exceed it	1e−9 tolerance comparison functions; thirteen boundary cases verified
Classification	Asset managers not isolated, exchanges wrongly isolated	Actual industry label values differed from assumption; pattern failed in both directions	Three layers: pattern, whitelist (18), blacklist (5)
Measurement basis	Forward growth showed +404%	Revenue reported net, forecast issued gross	Ratio and net-margin anomaly tests; field blanked and marked
Measurement basis	Mature industrial classified as early stage	Test omitted the "still small" component of the category's meaning	Scale gate added; runway capped at 40 quarters
Infrastructure	New feature showed zero matches	Edge cache lifetime header was also honoured by the browser	Separate headers per layer; browser copy marked non-storable
Observer	Preceding item took three rounds to diagnose	Test requests carried a timestamp, bypassing the very cache layer at fault	Retain one verification channel identical to real usage

---

The system described here is a personal research project. Its output is screened and organised data and does not constitute investment advice. Company tickers appear only as illustrations of method and are not recommendations. All thresholds and rules can be verified in the source code.

Thomas · September 2026