DG Fractional

FRACTIONAL CTO

A FREE TOOLKIT FOR FOUNDERS AND BOARDS

The Engineering Health Check

A twenty minute self assessment for founders, CEOs and boards who need to know whether their engineering function can carry the next stage of growth, before it becomes obvious that it cannot.

Org Design | Hiring and Retention | Delivery | Architecture | AI Adoption | Board and Investors

6	30	~20 min
Areas covered	Direct questions	To work through it

From David Goodman · DG Fractional

V1.0 · FREE EDITION

BEFORE YOU START

How to use this check

This is a compressed version of the same six area framework I work through in the first two weeks of an engagement. Org Design, Hiring and Retention, Delivery, Architecture, AI Adoption, and Board and Investors. It will not replace someone sitting with your team, but worked through honestly it takes about twenty minutes and tells you whether the way you run engineering is going to hold at the next size up.

1. Work through it with whoever runs engineering, or on your own first and then with them. The difference between your answers and theirs is usually the most useful part.
2. Answer honestly. This only works if nobody is marking their own homework. A vague or defensive answer to a direct question is itself a data point, so note it as a flag.
3. Tally the flags per area and use the scoring guide on page 6 to turn your answers into a plain verdict. Healthy, fixable this quarter, or get help now.

> **What a flag means here.** Not every no is serious. The same gap matters far less in a twelve person team than in a forty person one, and far less in a slow moving internal tool than in something customers depend on hourly. Weight every answer against what the business actually needs from engineering over the next twelve months, not in the abstract.

THE SIX AREAS, NEXT THREE PAGES

01 · Org Design

Is it clear who owns what?

> An engineering org that grew by adding people rather than by design is the most common reason delivery slows down after a raise. It almost never presents as an org problem. It presents as missed dates.

- [ ] Can every engineer name the one person who decides what their team works on next?
- [ ] Does any lead have more than eight or nine people reporting directly to them?
- [ ] Are teams built around parts of the product, or around who happened to be free at the time?
- [ ] Does ordinary work regularly need three or more teams to coordinate before anything ships?
- [ ] Is there a written definition of what a senior engineer is, that hiring and promotion both use?

RED FLAGS IN THIS CATEGORY / 5

02 · Hiring and Retention

Can you get good people in, and keep them?

> At this size, one senior departure can cost you a quarter. The question is not whether people leave. It is whether anyone sees it coming before the notice lands.

- [ ] Do you know how long it takes, on average, from opening a role to someone starting?
- [ ] Has an offer been turned down in the last six months, and does anyone know why?
- [ ] Is there a progression framework engineers can actually see, or is promotion a conversation?
- [ ] Can you name the three engineers whose exit would hurt most, and is anything reducing that risk?
- [ ] Does a new engineer ship something to production inside their first two weeks?

RED FLAGS IN THIS CATEGORY / 5

03 · Delivery

Do you know why things take as long as they do?

> Most teams measure output. Very few measure the waiting. The gap between work being ready and work being done is usually where the quarter quietly goes.

- [ ] How long does a typical change take from first commit to running in production?
- [ ] Do releases happen on a schedule, or whenever someone decides it feels safe enough?
- [ ] Is there an automated test suite that runs on every change, rather than just existing in the repo?
- [ ] When a date slips, does anyone establish why, or does the team simply move the date?
- [ ] What proportion of engineering time goes on unplanned work, and does anyone actually track it?

RED FLAGS IN THIS CATEGORY / 5

04 · Architecture

Built to grow, or just built to survive?

> The system works today. That is not the question. The question is what it costs you to put ten times the load, or three times the team, through it.

- [ ] Could the system handle ten times current load without a rewrite?
- [ ] Can the team name their single points of failure, unprompted?
- [ ] Could a new engineer reason about the data model, or does it require tribal knowledge?
- [ ] Are third party integrations contained, or wired through the whole codebase?
- [ ] Is there a part of the system nobody will touch without the one person who built it?

RED FLAGS IN THIS CATEGORY / 5

05 · AI Adoption

Is AI actually helping, or just present?

> Nearly every engineering team is now using AI tools. Very few have agreed what good looks like. The distance between those two facts is where the debt collects.

- [ ] Is there a written policy on what proprietary or customer data can go into third party AI tools?
- [ ] Is AI generated code reviewed by a human before merging, consistently rather than occasionally?
- [ ] Can anyone point to something measurably faster, rather than a general sense that things are?
- [ ] Do engineers actually use the tools, or were licences bought and never adopted?
- [ ] If AI is part of the product, can the team explain what happens when the model is wrong?

RED FLAGS IN THIS CATEGORY / 5

06 · Board and Investors

Does the board understand what it is being told?

> A board that cannot interrogate the engineering story eventually stops asking, which is worse. An absence of hard questions is not the same as confidence.

- [ ] Does the board get engineering numbers each month, or a narrative?
- [ ] Could your investors explain, in one sentence, what engineering is spending money on?
- [ ] When engineering asks for headcount, is the case made in business terms or technical ones?
- [ ] Has anyone independent looked at your engineering function in the last eighteen months?
- [ ] If technical diligence happened next quarter, do you already know what it would find?

RED FLAGS IN THIS CATEGORY / 5

TURNING ANSWERS INTO A VERDICT

What your flags actually mean

Count the no or flagged answers within each of the six areas, then read across. This is a blunt instrument on purpose. It is built to tell you when to stop and get a proper look, not to replace one.

0 to 1

Healthy

Nothing here should be holding you back. Revisit it after the next twenty hires.

2 to 3

Fixable this quarter

Real, but the sort of thing a focused piece of work sorts out before it starts costing you.

4+

Get help now

This area is already costing you money and time. It will not resolve itself as you grow.

One area outweighs the average. A company scoring well everywhere except Hiring, with four or more flags there, is in worse shape than one carrying two flags spread evenly. Concentrated weakness is what actually breaks, and it breaks at the least convenient moment. Do not sum the areas and average them out.

How an engagement differs. A checklist gives you a yes, a no, or a flag. In an engagement every finding carries three things. What it is costing you now, what fixing it takes in weeks and money, and whether it needs doing this quarter or can wait until after the next raise. That is what makes it a plan rather than an opinion.

YOUR SCORECARD, FILL IN AS YOU GO

Org Design	____ / 5	Hiring and Retention	____ / 5
Delivery	____ / 5	Architecture	____ / 5
AI Adoption	____ / 5	Board and Investors	____ / 5

Overall verdict ☐ Healthy ☐ Fixable this quarter ☐ Get help now

WHAT THE REAL THING LOOKS LIKE

Ninety days, one company

This check gets you a plain yes, no, or flag it. An engagement gets you the change itself. Here is what the first ninety days looked like at a company roughly the size of yours.

ENGINEERING, FIRST 90 DAYS

3 changes

```
engineering/delivery
- one release every six weeks, 40% rework
+ weekly release train, PRs reviewed same day
// two senior offers stalled, nobody owned hiring
```

Outcome. Cycle time halved by week ten. Both offers signed.

Nothing on that list is clever. A release train, a review expectation, and one person made accountable for hiring. The work was not identifying it. Most teams half know already. The work was having someone senior enough to make the call, present often enough to make it stick, and outside the politics enough to say it plainly.

That is the difference between a document telling you what to ask, and someone in the room for two or three days a month until the thing is actually fixed.

READ THIS BEFORE YOU RELY ON IT

What this check will not catch

A checklist tells you what to ask. It cannot sit in your stand ups, cannot tell whether an answer is accurate or merely optimistic, and cannot catch the things a team does not know to mention.

- ✕ It cannot tell you whether your test coverage is real or theatre. Tests that exist but assert nothing meaningful pass every check on page 4.
- ✕ It cannot tell you whether your senior engineers are staying because they are happy, or because they have not looked yet.
- ✕ It cannot tell you whether the architecture holds under the load you intend to put through it in eighteen months, only whether anyone has thought about it.
- ✕ It cannot put a number on any of it. What a weakness is costing you per quarter, and what fixing it takes in weeks and money.
- ✕ And it cannot do the part that actually matters, which is making the change stick after the conversation ends.

> That gap is the whole reason fractional leadership exists. Someone who has built and run engineering functions at this size looking at yours, testing what you have been told against what is actually happening, and then staying long enough to fix it. The next page covers what that looks like in practice.

ABOUT

Hands on senior engineering leader

I'm **David Goodman**. I've spent 20+ years in software engineering, the last 9 years leading teams of up to 50 engineers across fintech, cybersecurity and enterprise SaaS, from American Express through to fast moving scale ups and startups.

You're not hiring a firm that will send someone junior with a template. You're hiring someone who has run engineering functions at eight, twenty and fifty plus engineers, including taking a forty person function past fifty, and who will be the one in the room doing the work.

I've led AI transformation at organisation scale. Rolling out Claude Code and agentic development workflows, automated PR reviews and coding standards across whole engineering, data engineering and data science functions, rather than talking about it in a slide deck. My first degree was in Artificial Intelligence, back in 2006, with an MSc in Information Technology since.

> This check is the free edition of that same judgement, compressed into something you can run yourself in twenty minutes. The next page covers what happens when you want someone actually doing it.

WHAT COMES AFTER THE CHECK

Want someone actually doing it?

Four ways in, depending on where you are. A short review before you commit to anything, ongoing leadership on retainer, full time cover while you search, or a focused piece of work on AI. Every engagement is fixed fee, agreed in writing before any work starts.

ONGOING

Fractional CTO

Two or three days a month leading your engineering function. Org design, hiring, delivery, architecture direction and the technical story your board hears.

START HERE

Engineering Health Check

The full version of this document. A week looking at how your engineering function actually works, and what is worth fixing first.

AI

AI Adoption and Policy

The policy, the review process and the guardrails, then getting the tooling genuinely adopted rather than merely announced.

COVER

Interim Head of Engineering

Full time cover when the role is empty and the search will take months. I hold it together and help you hire the permanent person.

hello@dgfractional.co.uk · dgfractional.co.uk · linkedin.com/in/dgwebit

Based in the UK, working with companies wherever they are. Replies within one business day.