The Call for Agility

Chapter 1 — free, from Worth Building with Scrum: From Why to Done

Christian Neiderstam

Before you start

The project

Beauty Spot is a beauty retailer with around forty shops. Customers buy make-up and skincare over a counter, from staff who advise them. Their competitors have apps; they don't, and they're losing younger customers.

They're about to commission the **Branded Community App**, and this chapter is the first conversation about it. Nothing has been decided. Nobody has written anything down.

Who is in the room

Christian Neiderstam	*Scrum Master.* Brought in to run the process. The person telling you this.
Marilyn Moore	*Chief Executive.* Sceptical about anything that sounds like methodology. She's the one who has to sign.
Ben Hawkins	*Chief Marketing Officer.* Wants the app, and wants it now. He's also the one everybody finds difficult.
Alex Jensen	*Product Owner.* Newly appointed. Accountable for the product's value, and not yet sure what that means.

What to watch for

1. **The difference between a pressure and a driver.** Everyone in that room can name what's worrying them. Turning that into something you could check in a year is a different exercise, and almost nobody has been asked to do it.
2. **The moment the room goes quiet.** What a Scrum Master does in the next thirty seconds decides whether the drivers belong to the room or to him.
3. **Approval is not ownership.** The last section is the one to read twice. It's the difference between a stakeholder who nods and one who defends the work when it costs him something.

This is stage one of seven. Everything the team builds later traces back to what comes out of this room.

Chapter 1: The Call for Agility

The crossroads

The first time I sat down with Beauty Spot, nobody in the room used the word Scrum.

What they had was a problem. Marilyn Moore, the CEO, and Ben Hawkins, her CMO, wanted to build a Branded Community App — something that would change how the company met its customers, rather than just giving them another way to shop. They were excited about it. They had budget for it. And they had, between them, about nine different ideas of what it actually was.

That’s a good place to start, incidentally. Far better than a room where everyone already agrees, because agreement that early is usually just nobody having asked a hard question yet.

What Marilyn and Ben got right, and got right quickly, was something a lot of leadership teams never work out at all.

How they built it was going to matter as much as *what* they built.

Their market moved in weeks. Trends turned over between one campaign and the next, customers were being retrained by competitors who had been digital-first from the start, and nobody — not Marilyn, not Ben, not any consultant they could hire — actually knew what this app should be. They could make a good guess. They could not make a plan.

That’s the situation Scrum was built for. Not because it’s fashionable, and not because it makes teams faster, whatever you’ve been told. Because when you genuinely cannot know the answer up front, the only sensible thing to do is build a little, show someone, find out you were wrong about something, and adjust.

If Beauty Spot had been replacing a payroll system to a fixed regulatory deadline, I’d have told them to use something else. They weren’t. So we didn’t.

Enter the Scrum Master

This is where I stepped into the picture.

My role wasn’t tracking tasks or running meetings. It was guiding Beauty Spot through a change of habit:

- Embracing an agile mindset
- Building a culture of continuous improvement
- Turning bold strategy into something you could actually ship

The old way was top-down control, detailed upfront planning, rigid execution. The new way was servant leadership, iterative progress, team-driven decisions.

That shift doesn’t happen in a kickoff meeting. It happens in a hundred small moments, and the first one was about to arrive.

Laying the keel

Before you define a vision, you need something underneath it. An unshakeable *why*.

Shipbuilders lay the keel first. The spine that everything else is fastened to. Get it wrong and nothing above it sits straight. Business Drivers are that keel.

The rule behind the scene: Business Drivers

> A **Business Driver** is a strategic reason the organisation is doing this at all. Not a feature. Not a requirement. The commercial or operational pressure that makes the work worth funding.
>
> Two tests. A real driver survives the question *“and why does that matter?”* without collapsing into a feature request. And it can be paired with an **observable outcome** — something you could point at in a year and say whether it happened.
>
> Drivers are not a Scrum artifact. The Scrum Guide doesn’t mention them. I group them with Product Vision and Product Goal under a label of my own. The **Business Commitment Framework**, and I’ll be clear every time that the label is mine, not Scrum’s.

Ben was the right person to start with. As CMO he understood the market, the customer, and where the brand was heading.

So I went to his office with a whiteboard marker and one question.

“Ben, before we envision a single feature — what are the core business drivers this app has to serve for Beauty Spot to lead in the coming years?”

That question did a lot of work. Notice what it refuses to do. It doesn’t ask what the app should *have*. Ask a stakeholder that and you’ll get a feature list in ninety seconds, and you’ll spend the next six months building it without ever knowing why.

From pressure to driver

Ben started where strategists start, with the threat.

The market reality: Beauty Spot’s brick-and-mortar dominance was under attack from digital-native brands. The opportunity: a chance to redefine customer engagement. The threat: losing relevance entirely.

He leaned forward. *“Our traditional stronghold is under siege. We need to reclaim our edge.”*

There it was. I wrote it up:

Driver 1 — Enhance competitive positioning and market share

Then came the part most people skip.

“And what does success look like, Ben? If this works, what will you be able to point at?”

He thought about it properly. What we landed on was specific: a measurable increase in market share, a growing customer base, and e-commerce emerging as a dominant revenue channel.

That pairing — driver plus observable outcome — is what separates a strategy from a slogan. Do it for every driver. A driver you can’t measure is a driver you can’t inspect, and if you can’t inspect it you can’t adapt it.

The moment that decides whether you’re facilitating

With the first one down, ideas started flowing. And that’s exactly when it gets dangerous.

Ben lit up about live sales broadcasts — Beauty Spot experts demoing products on camera, customers buying inside the app.

“If we hit a million downloads, this could change the game.”

Now. He is the CMO. He is excited. He has just handed me a feature, not a driver, and it’s the kind of feature that eats a roadmap whole.

You have two bad options and one good one. You can say yes and lose the thread. You can say no and lose him. Or you can accept the energy and redirect the question.

“That’s a strong idea, and I want to come back to it. But before live sales can thrive — what makes the app indispensable from day one?”

Watch what that does. I didn’t judge his idea. I didn’t park it in a backlog he’d never see. I put it *after* something, which implies it has a place, and I handed the floor straight back to him.

His answer came instantly. *“Loyalty programmes. Exclusive in-store discounts, perks you can only get through the app. We need to make it essential.”*

Driver 4 — Strengthen brand loyalty and customer engagement

He got there himself. That’s the trick. A driver a stakeholder discovers is a driver a stakeholder defends.

Though I want to record something he said as we moved on, because it comes back.

“Fine. But we’re coming back to live sales, Christian. I don’t want to still be talking about loyalty barcodes in a year.”

I said of course we would. I meant it, and he didn’t quite believe me — and that gap, between a stakeholder who accepts a redirect and a stakeholder who trusts it, shapes a good deal of what follows in this book.

The full set

We worked the same way through the rest of the afternoon — pressure, driver, outcome, repeat — until we had five:

1. Enhance competitive positioning and market share
2. Increase revenue and e-commerce turnover
3. Improve marketing efficiency and reduce acquisition costs
4. Strengthen brand loyalty and customer engagement
5. Gather valuable customer insights, with consent

Five is about right. I’ve seen teams produce fifteen, and fifteen drivers means no drivers. Nothing is prioritised because everything is strategic. If you can’t hold them in your head, you’ll never use them to make a decision.

When the room can’t answer

Ben could do this. Plenty of stakeholders can’t, and it’s worth being honest about why.

It’s rarely that they haven’t got a strategy. It’s that nobody’s ever asked them to say it as an outcome you could check later. They know the pressures. Turning a pressure into *“we’ll know this worked when e-commerce turnover rises fifteen per cent”* is a translation job most executives have never been asked to do, and they won’t do it for the first time in front of an audience.

So you ask your good question, the room goes quiet, and now you’ve got a problem the Scrum Guide says nothing about.

Ask first. Always ask first.

Ben produced five drivers because somebody asked him a question worth answering. If I’d opened that session by handing him a list, I’d never have found that out. You don’t know what a stakeholder has until you ask, and most of them have more than they can get out in the first thirty seconds.

So ask the open question. Ask your Product Owner beforehand what he already knows, because he’s usually been in rooms you haven’t. Ask what the board’s been worrying about. What the last strategy day produced. What keeps coming up.

Then sit in the silence and let it be uncomfortable. Sometimes the answer turns up at second forty.

Only when it's clear the room genuinely can't produce it — not that it's slow, that it *can't* — do you reach for what I'm about to describe.

Here's what I do then. I generate a list of plausible business drivers for their industry beforehand. An hour with an AI model will do it. Then I keep it in my folder, unopened, in case I need it.

That's not a shortcut. It's the same trick as the impact-and-effort matrix in Chapter 3, and it works for the same reason: **people are far better at correcting a wrong list than producing a right one from nothing.** *"No, that's not it, ours is more about—"* gets said in seconds. *"What are your business drivers?"* can produce ten minutes of nothing.

There's a side effect worth knowing about. Having the list in your folder makes you a calmer facilitator. A Scrum Master with no fallback fills the silence at about twenty seconds, because it's unbearable and he'd rather be seen doing something. A Scrum Master with a prepared list can wait a full minute without panicking. And in that extra forty seconds, the room quite often answers on its own.

So the list often does its job without ever leaving the folder.

> **The trap, and it's a real one.** `[NEIDERSTAM]`
>
> A stakeholder handed a plausible list will sometimes just take it. Now you've got five professional-sounding drivers that belong to nobody, which is worse than having none. They look finished. Nobody will question them for a year.

> The generated list isn't an answer. It's bait.

What makes the difference is what you attach to each candidate: three or four questions the stakeholder can't answer from his chair.

Not *"is market share important to you?"* That's a yes, and it costs nothing. Something more like: *which competitor took business from you last year, and what did they offer that you didn't?* He won't know offhand. He'll have to go and find out, or ring somebody, or think properly about it for the first time in a while.

That's the whole point. **Ownership doesn't come from agreeing with a driver. It comes from the work of qualifying it.** A CMO who spent two days establishing that acquisition cost is the real pressure, and that market share was a symptom of it, owns that driver in a way no amount of nodding in a workshop can manufacture.

Keep it to three or four questions each. More becomes homework, homework gets delegated to somebody junior, and then your drivers are owned by an analyst instead of a CMO. Which is where you started.

The pressure that isn't a driver

There's a failure mode here that costs real money, and I've watched it happen more than once.

What a stakeholder feels most sharply is usually the most recent thing. Sales are down this quarter. A competitor launched something in March. A distributor changed terms. All true, all urgent, and often **symptoms of something structural nobody in the**

room has named — because everybody in the room is inside it.

Take the symptom as a driver and you'll fund the wrong thing with total conviction. You build an app to win back customers who aren't choosing a competitor at all. They're buying the category differently, or less. The app ships, meets every criterion, and the number doesn't move. Now you've spent a year proving the driver was wrong.

The correction is cheap. **Bring evidence from outside the building.** Sector reports, trade association data, analyst notes, published figures from listed competitors. Feed those in alongside what the room told you. What comes back is usually a driver or two nobody mentioned, because they're too slow-moving to feel urgent and too big to see from inside.

> **Drivers and blockers aren't the same thing.** `[NEIDERSTAM]`
>
> A **business driver** is a long-term outcome worth investing in. It survives a change of quarter. Money spent against it compounds.
>
> A **business blocker** is something in the way right now. A broken process, a contract, a system nobody can replace this year, a team that can't hire. It's real, it hurts, and it's almost never fixed by building software.
>
> Both arrive in the same sentence, in the same tone of voice. Telling them apart is most of what a good analyst is for, and it's worth doing before you write the drivers down rather than after.

> Put the blockers on a list of their own and hand them back to the business. Some are somebody else’s job. Some are a project. One or two will turn out to be the actual driver in disguise, and finding that out early is worth the whole exercise.

The prompt I use, and the shape of the qualifying questions, is on the board with the drivers. Take it and make it yours. The industry knowledge in it matters far less than the questions do.

Approval is not ownership

We had the drivers. We did not yet have the thing that actually mattered.

“Ben, this is a strong framework. Next step is Marilyn, but not as a decision to sign off. As a structured reflection of what she already said.”

I gave him the framing to use:

“Marilyn, reflecting on your strategic overview from the last board meeting — do these draft Business Drivers accurately capture the priorities you outlined?”

Read it once more, because the wording is the entire technique.

It doesn’t ask her to approve our work. It asks her whether we listened correctly. She isn’t a gate; she’s the source. And a CEO answering that question is not signing off on a document. She’s editing her own words.

Days later Ben was ecstatic. *“Christian, Marilyn loved it! She refined a few points, made them her own, and now they’re locked in.”*

She refined a few points. Perfect. Those edits are what converted approval into ownership.

Then, grinning: *“And I guess we’ve just made our first board item you keep talking about.”*

I smiled. Agile was taking root, not just in the process, but in the mindset.

Why this matters more than it looks

Here’s what happens when you skip this.

Leadership approves the drivers without touching them. Six months in, something goes wrong. And because they only ever approved it, they can step back from it — *we supported the initiative, but the delivery didn’t meet expectations.* Suddenly the Product Owner is standing alone holding a strategy nobody else will claim.

I’ve watched it happen. It is the single most common way an agile transformation dies quietly.

Marilyn didn’t approve those drivers. She rewrote two of them. That makes them hers, and it means she will defend them at a board meeting when I’m not in the room.

> **Plan B, if the ownership hadn’t come.** Had I sensed hesitation, I’d have called an alignment workshop, not as damage control, but as a structured way to re-engage leadership: audit the drivers together, let them shape the final

> wording themselves, and start the conversation about why their ownership decides the outcome. If leadership won't own the *why*, there is no point refining the *what*.

I added Marilyn to the board directly. Her reply came back fast:

"Ben, thank you — I'll bring this to the next board meeting. It reinforces that the challenges I outlined weren't just words; we're actively shaping solutions. I understand you'll be documenting progress alongside Christian, and I saw the mention of defining a vision for the platform. Excited to see what's next."

That was music to Ben's ears, and to mine.

We hadn't just written down strategic imperatives. We'd secured an executive champion.

The keel was laid. Next, we'd chart the course. The Product Vision.

Key takeaways

1. **Drivers before features.** Ask what the business needs to achieve, never what the app should have. The second question answers itself later; the first one never gets asked at all if you skip it.
2. **A driver without a measurable outcome is a slogan.** Pair every one with something you could point at in a year.
3. **Redirect, don't reject.** When a stakeholder hands you a feature, put it *after* something rather than declining it. You keep the idea, the energy, and the person.

4. **Ask leadership whether you listened, not whether they approve.** Their edits are what turn approval into ownership.
5. **Ownership is the insurance policy.** Approval lets leadership step back when things go wrong. Ownership means they're standing next to you.
6. **Five drivers, not fifteen.** If you can't hold them in your head, you can't use them to decide anything.
7. **Ask before you prepare anything.** The stakeholder usually has more than they can say in thirty seconds. Open with a list and you never find out what they had.
8. **A wrong list beats an empty room.** When they genuinely cannot produce drivers, candidates to argue with work better than a question to answer. Held in reserve, not opened with.
9. **Qualify with questions they can't answer from their chair.** Ownership comes from the work of establishing a driver, not from agreeing to one.
10. **The most urgent pressure is often a symptom.** Bring evidence from outside the building, or you'll back the wrong work and be certain you were right to.
11. **A blocker is not a driver.** One is a long-term outcome worth investing in; the other is something in the way that software rarely fixes. Separate them before you write anything down.

The artifact from this chapter

How to facilitate the right business drivers out of your own room

That afternoon with Ben is written up as something you can use on Monday. **It's the hardest part of a Scrum Master's job and the part no certification covers** — getting outcomes out of people who have pressures rather than drivers, and getting them stated so the room owns them rather than you.

The homework to do before you walk in. A research prompt for understanding how the client got here. The driver discovery prompt itself. What a qualifying question has to do to be worth asking, and why at least one of them must be able to *disqualify* the driver.

With a worked example: the six candidates it produces for Beauty Spot, and what the room would have done with each.

Free, no sign-up, no email — whether you buy the book or not.

neiderstam.com/artifacts/driver-discovery

That was one chapter of twenty-two

The drivers are stage one. The other six — vision, Product Goal, capability areas, Epics, stories, acceptance criteria — each answer the question above them, and each has its own artifact you can copy.

Then the team starts building, and the book follows them through four Sprints to find out whether any of it moved.

Worth Building with Scrum: From Why to Done

$19.99 · PDF and EPUB · 244 pages · paperback on Amazon

neiderstam.com
