



BASE REALITY GROUP

COMPANION GUIDE

The Operator's Guide to *Product Leadership*

Every worksheet, test and chart from the twelve-part course, one to a page, for the person doing the head-of-product job.

The job, *on paper*

The Operator's Guide to Product Leadership is a twelve-part course built on one idea: the product job is four decisions. This guide collects every worksheet, test and chart the course uses, one to a page, so you can make those decisions on paper, about your own company.

It's for heads of product, and for the people doing the job without the title: a founder or CEO still holding product, a first-time head of product, a senior product manager moving into leadership, an engineering or commercial leader who inherited product because nobody else had it. You don't need a product team, a strategy document or any particular vocabulary to use it. The course explains every system before it uses one.

The four decisions are which problems you solve and for whom, what evidence justifies solving them, what you decline, and what outcome you expect and how you'd know. Every company that ships anything answers them, whether or not anyone wrote them down. The course asks whether someone answers them on purpose, and nearly every page in this guide either finds out who answers one of them today or helps you answer it deliberately.

The pages follow the course, module by module. Each module opens with a short summary and a link to the video, then gives each tool its own page: what it's for, how to use it, and the sheet itself. Print them, write on them, and keep the filled-in pages together, because later modules build on earlier ones. The audit from module 2 comes back in modules 3 and 12, the decline from module 5 gets its escalation rule in module 6, and the bar from module 7 becomes the scoreboard in module 8.

How the pages are written

Meridian is a composite. The course follows a hypothetical forty-person company with about sixty customers, six engineers, one designer and a founder who holds product. Where a page shows Meridian's answer, read it as an example of a filled-in line. Its numbers belong to the example.

Every page says where its idea came from. Some tools are the course's own construction, and the page says so. Others come from named practitioners, among them Marty Cagan, Roger Martin, Teresa Torres and John Cutler, and the page names them. Several of those sources sell training, coaching or consulting. Where the course says so, the page does too.

Most of these tools have no study behind them. Where the course looked for evidence that a practice works and found none, the page says that too. What the tools give you is mechanical: a decision written down can be pointed at, a bar written before the count can be checked against the count, and a date on the shared calendar reminds people in a way a page can't.

Most pages take minutes. Module 2's audit takes about fifteen. Meridian's founder wrote the focus line of his one-page strategy in about ten minutes. A few pages are logs you keep for a quarter, and several are meant to be filled in again: module 12 ends with Meridian's founder telling his board he'll fill in his page again at the next strategy refresh.

What's in this guide

01 Your Company Already Has a Head of Product	5	07 Your Biggest Customer Wants It. Does Anyone Else?	32
The four decisions, on one page	6	Supply and demand	33
Hold your roadmap up against the four	7	The four big risks	34
02 Four People Are Deciding Your Product Strategy	8	Set the bar first	35
The audit	9	Ask about the past	36
The shadow org	10	A pace a forty-person company can keep	37
03 You Gave Them the Title. You Kept the Decisions.	11	08 You Shipped It on Time. Nobody Asked If It Worked.	38
Executive altitude and execution altitude	12	The scoreboard, written before anyone knows the score	39
Find your last real no	13	O'Reilly's hypothesis template	40
The transfers that hold	14	The column after Done	41
Keep them, deliberately	15	Reporting the miss	42
04 You Told Them to Own It. Then You Took It Back.	16	09 You Hired Someone Good. Then You Answered for Them.	43
The product operating model	17	One page for a new hire	44
Scored against the five	18	Five levels, and how to coach at each	45
Clarity and competence	19	The weekly one-on-one	46
Ten ways companies get this wrong	20	Gap analysis	47
The ideas, and the machinery	21	10 Your Head of Sales Blamed Product	48
05 They Asked for Your Strategy. You Opened the Roadmap.	22	The sheet	49
Three things on one page	23	When the CEO wants an exception	50
A plan, or a strategy?	24	No surprises: the board	51
The last thirty decisions	25	11 You're Planning Next Quarter. You Never Checked This One.	52
06 Your Biggest Customer Asked for the Thing You Said No To	26	The quarter on one page	53
RICE, scored twice	27	Holding a decision until its date	54
Outcomes you owe	28	12 Your Board Wants a Chief Product Officer	55
The escalation rule	29	Held on purpose, or by default?	56
When the request comes in: one phone call	30	Who takes over a decision	57
What did the decline buy?	31	The handover	58

Your Company Already Has a Head of Product. Nobody Told Them.



What decisions does this role own that no other seat owns?

The head-of-product job is easiest to define by its decisions, because the activities keep changing and the decisions hold still. Module 1 names four: which problems we solve and for whom, what evidence justifies solving them, what we decline, and what outcome we expect and how we'd know. Roadmaps, specs and standups are what the work leaves behind, and all of them can exist in a company where the decisions never got made.

At Meridian, the course's composite company, the four get answered anyway. The latest customer escalation decides which problems get solved, nobody sets the evidence bar, engineering capacity decides the declines around week seven of each quarter, and nobody ever states an outcome.

WATCH MODULE 1 youtu.be/BxT0oNf9BpM

THE PAGES FOR THIS MODULE

The four decisions, on one page	6
Hold your roadmap up against the four	7

The four decisions, on one page

The course's definition of the job, tested against the last thing your company shipped. The four describe something already happening: your company answers them every time it ships, whether or not anyone has written them on a wall.

HOW TO USE IT

1. Give it about ten minutes, and do it for your own company.
2. Take the last significant thing you shipped. For each decision, write what the answer was for that release, as well as you can reconstruct it.
3. Don't try to answer the four properly yet. Write what was true, and leave a row blank where nobody decided anything.
4. Expect the third row, the declines, to be the hardest. The module predicts you won't be able to name a single thing your company has said no to on purpose.

DECISION	THE QUESTION BEHIND IT	FOR THE LAST THING YOU SHIPPED
Which problems we solve, and for whom	Out of everything a customer could want from you, which slice are you in the business of fixing, and for which customer?	
What evidence justifies solving them	Before you spend four engineering months on something, what would you need to have seen to believe it's worth four engineering months?	
What we decline	What are you saying no to on purpose, and expecting to still be saying no to a year from now?	
What outcome we expect, and how we'd know	Before you build it, what has to be different afterward for it to have been worth doing, and what would you look at to find out?	

Evidence and a reason are different things. One angry email from your largest account is a reason. "Eleven of our sixty customers asked for this unprompted, and four described the same workaround" is evidence. With eight customers instead of sixty, lower the bar and say so out loud: somebody set it on purpose.

The four are the course's own formulation, from practice. They don't come from a textbook.

From module 1, Your Company Already Has a Head of Product: youtu.be/BxT0oNf9BpM



Hold your roadmap up against the four

A roadmap records what’s being built, when, and by whom. Its job is coordination, and it does that well. The trouble is that it looks like the product job, so this test checks which of the four decisions it records.

HOW TO USE IT

- 1. Put your current roadmap next to this page.
- 2. For each decision, mark whether the roadmap says anything about it.
- 3. Where it doesn't, write where that decision gets answered today. Module 2’s audit is the full version of this question.

DECISION	DOES THE ROADMAP SAY?	WHERE IT GETS ANSWERED TODAY
Which problems we solve, and for whom	☐ Yes ☐ No	
What evidence justifies solving them	☐ Yes ☐ No	
What we decline	☐ Yes ☐ No	
What outcome we expect, and how we’d know	☐ Yes ☐ No	

Meridian’s roadmap, from the module: four quarters across, color coded, every initiative with an owner and a date, and silent on all four decisions. Every item’s definition of done was that the thing shipped.

The test, like the four decisions, is the course’s own.

From module 1, Your Company Already Has a Head of Product: youtu.be/BxT0oNf9BpM

Four People Are Deciding Your Product Strategy. None of Them Know It.



In a company with no head of product, where do these decisions get made?

A decision without an owner still gets answered, by whichever constraint pushes hardest and soonest. Module 2 calls the result the shadow org: escalation owns which problems, nobody owns the evidence, capacity owns the declines, and silence owns the outcomes.

At Meridian, a dispatcher at a customer in Ohio decides which problems get solved, nobody sets the evidence bar, the lead engineer decides the declines, and a roadmap column called Done stands in for the outcome, all while the founder tells people he owns product. The dispatcher, the lead engineer and the founder are each right from the one window they can see through, which is why the module calls its audit “a diagnostic and not an accusation.”

WATCH MODULE 2 youtu.be/erp1LpqROWE

THE PAGES FOR THIS MODULE

The audit	9
The shadow org	10

The audit

Before anyone reassigns a decision, find out who makes it today. A department never decided anything, and a process never decided anything either, so this page puts a name beside each of the four: a person, a customer, or nobody.

HOW TO USE IT

- 1. Allow about fifteen minutes. Beside each decision, write who makes it: a real person in your company, a customer, or the word *nobody* if that's the truth. Never a department.
- 2. If what you find is a spreadsheet column or the calendar, write that down accurately and don't fix it yet. You can't reassign a decision you haven't located.
- 3. In the third column, write what would have to be true for that name to change. Leave who should own it instead for a later, bigger conversation.
- 4. Notice which row took longest. In the module's experience, it's the one that has been running on its own the longest.

DECISION	WHO MAKES THIS CALL TODAY	WHAT WOULD HAVE TO BE TRUE FOR THAT NAME TO CHANGE
Which problems we solve, and for whom		
What evidence justifies solving them		
What we decline		
What outcome we expect, and how we'd know		

1 Which of the four took you longest to name?

The third column can be small. For the evidence row, it might be somebody writing down what counts as enough. For the decline row, a single sentence somebody is willing to say out loud in a sales meeting.

The module's view is that every name on this page belongs to somebody doing their job well under the constraints they can see.

From module 2, Four People Are Deciding Your Product Strategy: youtu.be/erp1LpqROWE

The shadow org

Who answers each decision when nobody has been given it. Read it with your own company in mind.

HOW TO USE IT

- 1. Read each row and mark whether it describes how the decision gets answered at your company.
- 2. Carry anything you mark into the audit on the previous page, with a real name.

DECISION	ANSWERED BY	HOW IT SHOWS UP	IS THIS US?
Which problems	Escalation	Whoever escalated most recently decides, whatever their size or strategic value.	☐ Yes ☐ No
What evidence	Nobody	Nobody decides how much proof is enough, so whoever asked last sets the bar, and it moves every week without anyone noticing.	☐ Yes ☐ No
What we decline	Capacity	The team runs out of hours around week seven of a quarter, and whatever hasn't started is effectively declined. The calendar said no.	☐ Yes ☐ No
What outcome	Silence	Nothing that ships is ever measured against anything anybody committed to in advance.	☐ Yes ☐ No

The module calls the shadow org a company being responsive, which is a virtue. What's missing is anybody holding the four questions in their head at the same time.

You Gave Them the Title. You Kept the Decisions.



What is the difference between a CPO or VP seat and a director or senior manager running product?

Meridian’s new head of product declines a prospect’s custom integration, and by Thursday it’s back on the roadmap, because the account executive took it to the founder in a hallway. Module 3 separates skill from authority. You can hire the skill. Authority gets confirmed in public, every time somebody tries to route around the person who holds it.

Drawing a line through the four decisions gives two altitudes. Executive altitude owns which problems the company solves and what it declines; execution altitude owns that the chosen problems get solved well. The test of whether authority landed is the decline, and keeping the decisions yourself, deliberately, is a legitimate answer too.

WATCH MODULE 3 youtu.be/IXP_vO7acso

THE PAGES FOR THIS MODULE

Executive altitude and execution altitude	12
Find your last real no	13
The transfers that hold	14
Keep them, deliberately	15

Executive altitude and execution altitude

Draw a line through the four decisions between the two that set the company’s direction and the two that make sure the direction gets delivered. The first pair is executive altitude, the second execution altitude: same craft, different decisions.

HOW TO USE IT

- 1. Copy who holds each decision today from your audit (module 2).
- 2. Mark whether each one is held on purpose.
- 3. If nobody holds the executive pair on purpose, the seat you need is an executive one, whatever title you planned to hire. At Meridian a customer’s escalations and an engineer’s capacity held that pair.

DECISION	ALTITUDE	WHO HOLDS IT TODAY	ON PURPOSE?
Which problems we solve, and for whom	Executive		☐ Yes ☐ No
What we decline	Executive		☐ Yes ☐ No
What evidence justifies solving them	Execution		☐ Yes ☐ No
What outcome we expect, and how we’d know	Execution		☐ Yes ☐ No

The tell. You can call someone a chief product officer, but if the decision about what the company won’t do still routes to the CEO, the company has hired execution altitude: a VP of product with a CPO title.

The split is the course’s own framing; you won’t find it in a textbook. Which problems and what we decline are the same decision seen from two sides, because every yes is a no to something you didn’t pick.

Find your last real no

The test of whether a product leader's authority landed is the decline. Anybody can be handed permission to say yes, and it costs them nothing. A no has a name attached to it, because somebody wanted that thing.

HOW TO USE IT

1. Find the last time your company declined something a customer wanted, and write it down with when it happened.
2. Check it against the three marks of a real decline.
3. Write who made the call.
4. Read the answer. If it was you, that's fine, as long as you meant it. If it was you and somebody in the building has head of product on their business card, those two facts are in conflict, and the module's view is that the person on the card has known for a while.

The decline

WHAT A REAL DECLINE LOOKS LIKE

- ☐ It names the thing.
- ☐ It names who wanted it.
- ☐ It survived longer than one conversation.

Who made the call

When it was made

The module's example of a real decline: "We're not building custom integrations for single customers this year, and here is where that came from." Somebody can be angry about it and it still stands.

If you can't think of a decline like that from the last six months, the module reads it as a company where nothing has been declined on purpose, which it calls more urgent than any argument about titles.



The transfers that hold

Authority arrives in two pieces. It's granted once, on the day someone writes the job description, and then confirmed in public every time somebody tries to route around the new owner. You can hire the skill, but the confirmation has to happen in front of everyone, so this page plans both acts for one decision.

HOW TO USE IT

- 1. Name the decision in a sentence a person can repeat back. The module's example: *She decides what we decline this quarter.* That sentence is the transfer.
- 2. Say the same sentence to everyone who would otherwise route around the new owner. At Meridian that would mean the account executive, the engineers and whoever runs customer success.
- 3. When the first day comes that the transfer costs you something, back it, and write down what happened.

The decision being handed over

The sentence, in words a person can repeat back

WHO NEEDS TO HEAR IT	TOLD ON

The first day it cost you something: what happened, and did the decision stand?

The module puts the handover conversation at about ninety seconds. It's simple, and it's uncomfortable.

Keep them, deliberately

Keeping the decisions yourself is a legitimate answer too. If you're forty people and the direction of the company is the thing you're best at, keeping the problem and decline decisions for yourself is defensible. Keeping them by accident, then hiring somebody to be frustrated by them, is not.

HOW TO USE IT

- 1. Take about fifteen minutes. Mark each of the four: you keep it, or it goes to the person you're hiring.
- 2. Tell the person you're hiring which ones you're keeping, before they say yes.

DECISION	I KEEP IT	IT GOES TO THE HIRE
Which problems we solve, and for whom	☐	☐
What evidence justifies solving them	☐	☐
What we decline	☐	☐
What outcome we expect, and how we'd know	☐	☐

Who I'm hiring, and for which role

Date I told them which decisions I'm keeping (before they accepted)

Module 12 comes back to this page with a test for whether a kept decision is held on purpose.

You Told Them to Own It. Then You Took It Back.



Is there a known system for this, and what does it give you?

Module 4 teaches the product operating model, mostly from Marty Cagan and the Silicon Valley Product Group: three dimensions, five concepts, and a durable team given problems instead of features and held to outcomes instead of output. It also explains why “you own this” so often comes back to your desk. In product coach Marcus Castenfors’ formulation, empowerment only works once the person already has clarity and competence.

Read against a forty-person company, the model splits into ideas, which are cheap and can start on Monday, and machinery, which needs an organization the company doesn’t have. There’s no independent evidence the model works, the module adds, and everyone it cites sells transformation services.

WATCH MODULE 4 youtu.be/LCXgYDG7XUk

THE PAGES FOR THIS MODULE

The product operating model	17
Scored against the five	18
Clarity and competence	19
Ten ways companies get this wrong	20
The ideas, and the machinery	21

The product operating model

A named, end-to-end system for product work, mostly from Marty Cagan and the Silicon Valley Product Group. The course teaches it because the rest of the series uses its vocabulary, and then asks who it was written for.

HOW TO USE IT

1. Use it as a glossary for the modules that follow.
2. Then score your own company against the five concepts on the next page.

Three dimensions

How you build. How you solve problems. How you decide which problems are worth solving at all.

The leader's three parts

Lead with context instead of control. Build the team's competence by coaching them. Make sure they can solve the problem you gave them.

THE FIVE CONCEPTS

CONCEPT	WHAT IT MEANS, AS THE MODULE TEACHES IT
Product culture	What the company rewards when nobody's writing it down. To find it, look at who got promoted last.
Product strategy	Choosing what you'll be good at, and therefore what you'll be bad at, and saying so where your team can hear it.
Product teams	A small team that stays together, with an engineer, a designer and a product manager, given a problem to solve.
Product discovery	The work before you commit engineering months, to find out whether the thing is worth building at all.
Product delivery	Getting software into customers' hands often and safely, without it turning into an event every time.

Feature team or empowered team

A feature team is handed the answer and asked to build it. An empowered team is handed the problem. From outside, on any given morning, you can't tell them apart.

Output or outcomes

If you're accountable for output, you're done when it ships. If you're accountable for outcomes, shipping is the middle of the job.

The module found no independent evidence that the model works and no checkable number for how many companies run it or how many transformations succeed. Everyone it cites sells transformation services, including the people who publish warnings about the model.

Scored against the five

Put your company against each of the model's five concepts, in words, the way the module scores Meridian. Meridian came out strong on delivery, absent on discovery, and not yet applicable on teams.

HOW TO USE IT

1. For each concept, write a plain sentence or two about where your company is.
2. Use Meridian's column to calibrate the kind of sentence you write.
3. If a concept doesn't apply yet at your size, say so. Teams didn't apply at Meridian.

CONCEPT	MERIDIAN, AS THE MODULE SCORES IT	YOUR COMPANY
Delivery	Strong. Six engineers who ship every week, almost never break anything, and can put a fix in front of a customer the same afternoon.	
Discovery	Zero. In four years nobody ran a single test to find out whether something was worth building before building it.	
Strategy	Not written down. The founder could tell you over lunch; ask the six engineers and you'd get six answers, none of them quite his.	
Teams	Doesn't apply yet. Six engineers, one designer and a founder who decides.	
Culture	Good and trusting; nobody's afraid. But it has never been asked to make a decision explicitly, so nobody knows what it does under pressure.	

The five concepts are the product operating model's, from Marty Cagan and the Silicon Valley Product Group. Scoring them in words is the module's.

From module 4, You Told Them to Own It. Then You Took It Back.: youtu.be/LCXgYDG7XUk



Clarity and competence

In Marcus Castenfors' formulation, empowerment only becomes possible once two things are already true. If either is missing, "you own this" creates a person alone in a room with a problem they can't size and no sense of direction.

HOW TO USE IT

1. Before you hand someone a decision, answer both questions about them.
2. Then look back. Find the last decision you handed somebody and then took back. Don't apologize for it; just work out which of the two was missing.
3. Watch for the loop that brings a decision home: they go quiet, you ask for an update, you ask again, and you make the call yourself.

- | | |
|--|--|
| 1 Clarity. Does this person know the direction of the product: what you're trying to become, and what you've decided you're not? | ☐ Yes ☐ No |
| 2 Competence. Can they solve the problem you just handed them? Have they done it before? Can they talk to a customer without leading the witness, or tell a real signal from three loud emails? | ☐ Yes ☐ No |

The last decision you handed over and took back

WHICH OF THE TWO WAS MISSING?

- ☐ Clarity was missing.
- ☐ Competence was missing.

THEN BUILD THE TWO PILLARS

- 1** Write the direction down, so everyone on the team would say the same sentence.
- 2** Decide which problems each person can take.
- 3** Coach the gap you find.

Castenfors, a product coach who sells transformation work, walked through this on the Silicon Valley Product Group's podcast. It's practitioner experience, not a study.



Ten ways companies get this wrong

Marcus Castenfors' field guide lists ten ways companies go wrong adopting the model. Read against a forty-person company, five of the ten need parts it doesn't have, which makes the list, in the module's words, "a list of prerequisites, wearing a warning label." The other five can happen at any size.

HOW TO USE IT

1. Check the first group against your company. Each of these can happen at forty people.
2. Read the second group as prerequisites. Check one only if the piece it worries about is already in your building; otherwise it doesn't apply yet.

AVAILABLE AT ANY SIZE: CHECK THE ONES YOU RECOGNIZE

- ☐ Giving teams too much autonomy too soon.
- ☐ Never balancing new work against keeping the lights on.
- ☐ Letting the change stall unseen.
- ☐ Answering the what and never the why.
- ☐ Leading without changing how you lead.

NEED A PRODUCT ORGANIZATION YOU MAY NOT HAVE YET

- ☐ Running the change inside the product and technology bubble, and leaving the rest of the business in the dark.
- ☐ Unmanaged dependencies between teams.
- ☐ A chief product officer and a chief technology officer who haven't built a partnership.
- ☐ Bringing design in after the decisions are made.
- ☐ Going all in instead of starting with pilot teams.

The module numbers only some of the ten, so they're unnumbered here; the first group's wording follows the module's companion article. This is practitioner field experience, not a study, and Castenfors sells transformation work.

The ideas, and the machinery

The model comes in two piles with different price tags. Its machinery needs an organization you don't have and costs money, so shelve it for now. Its ideas are cheap and can start on Monday.

HOW TO USE IT

- 1. Write the date you start each idea.
- 2. Leave the machinery shelved. It answers a question about coordinating dozens of teams, and a forty-person company has one team.

THE IDEAS: START ON MONDAY

THE IDEA	WHAT IT COSTS	STARTED
Hand somebody a problem instead of a solution.	One conversation, and the discipline to stop talking before you describe how you'd have done it.	
Find out something is a bad idea before you build it.	A week of somebody's time and a prototype that doesn't work.	
Say what you expect to happen when a thing ships, write it where you'll find it, and go back in six weeks and look.	Twenty minutes, twice. The going back turns a guess into something you know.	
Build the two pillars: clarity and competence.	Time, more than it sounds. You build competence every time you leave a decision with the person you gave it to.	

The machinery, shelved for now: pilot teams, a coaching council to support them, a transformation office to run the whole thing, and a chief product officer and chief technology officer who agree with each other.

Separating the two piles is the course's own move. What John Cutler sees working, paraphrased in the module, is companies that start from a model and decide deliberately what to take and what to change.

They Asked for Your Strategy. You Opened the Roadmap.

What makes a product strategy real rather than a slide?

A roadmap is a list of things you control, so everyone in the room can agree to it, and it can say when things will happen but never whether they should. Following Roger Martin, a plan is a list of activities and a strategy is a theory about how you win. The sources the module reads agree on three things: a strategy is a choice, a choice means giving something up, and a strategy should fit on one page.

A company with nothing written down still has a strategy, readable off its last thirty decisions. Meridian's, the module says, was written by its inbox. The one page that replaces it has three lines: the focus, the insight, and the decline, the sentence that names what you won't do.

WATCH MODULE 5 youtu.be/uerj3F4aoyo

THE PAGES FOR THIS MODULE

Three things on one page	23
A plan, or a strategy?	24
The last thirty decisions	25

Three things on one page

A strategy on one page, with three lines on it. A mission statement, the module says, survives unchanged for a decade because nobody can act on it or violate it. This is the opposite kind of document: one you could break by Thursday.

HOW TO USE IT

1. Write **the focus**: who you're for. The module finds it's rarely the hard part.
2. Write **the insight**: what you understand about those customers that a bigger competitor doesn't. A focus with no reason under it is a preference, and preferences lose arguments.
3. Write **the decline**: what you won't do, by name, in a sentence somebody could hold you to. Start it with *we do not* and finish it. If you can't finish it, that's your finding.

1. The focus: who we're for

2. The insight: what we understand about them that a bigger competitor doesn't

3. The decline: we do not ...

Meridian's page. Focus: mid-size logistics operators running twenty to two hundred vehicles. Insight: a twenty-vehicle operator has no IT department, so what wins is a product that works on the first day, with no project attached; the enterprise vendors' business model assumes a six-week implementation. Decline: we don't build custom integrations for enterprise warehouse systems.

The insight might be wrong, and that's what makes it a strategy: it's the thing you'd change your mind about if the market told you otherwise. The decline is the line the module couldn't find in any source it read, and the one that costs real revenue. No study shows a written decline makes a company more successful; what it gives you is a document that forbids something, and you can settle an argument with that.

A plan, or a strategy?

Asked for a strategy under pressure, the module says, you produce the document that doesn't frighten you, which is the roadmap. These three checks tell a plan from a strategy.

HOW TO USE IT

1. Hold your current strategy document, or your roadmap, against the three questions.
2. A no on any of them means you're holding a plan. A plan does a real job, but it can only answer when something will happen, never whether it should.

- | | | |
|----------|--|--|
| 1 | Who decides whether you were right? A plan lists inputs you control, and you can tick off every line and still lose. A strategy names an outcome that depends on customers, who don't work for you. Does your document name an outcome your customers decide? | ☐ Yes ☐ No |
| 2 | Does it turn anything down? Everybody can agree to a list of things you're going to do, because nobody has to give anything up to say yes. Does your document refuse anything? | ☐ Yes ☐ No |
| 3 | Does it make you anxious? Roger Martin's point, paraphrased: you can't prove a real strategy in advance, and you have to commit to it anyway. Does your document make you anxious? | ☐ Yes ☐ No |

Martin's full method is five choices that all have to reinforce each other: what winning looks like, where you play, how you win, what capabilities that needs, and the management systems that keep it alive. The module points out that it assumes management systems. At forty people you have a founder, a spreadsheet and a group chat, and one page is the part of the method that still works without them.

Martin's material is paraphrased in the module. Felix Oberholzer-Gee defines strategy differently, as a plan to create value. The module leaves the two definitions standing side by side, because blending them would produce a definition neither man wrote.

The last thirty decisions

A strategy you never wrote down still leaves a pattern, and the pattern is sitting in your last thirty decisions. Read it before you write a new one. The module calls the diagnosis free: it takes an afternoon.

HOW TO USE IT

- 1. Look at your set of products and ask the module’s question first: *how did this happen?* If they grew from one idea and drifted apart, there was a coherent idea once, and your job is to find it again. If they arrived separately, there may be no shared spine, and writing one anyway gets you a document nobody can execute.
- 2. List the last thirty decisions of any size: what you built, what you didn’t build, and who you said yes to. Mark each B (built), N (not built) or Y (a yes to a customer or a deal, with who). A decision can carry both B and Y.
- 3. Write the pattern down as though it were a strategy somebody else handed you.

How did this happen? One idea that drifted, or products that arrived separately?

#	DECISION, AND WHO IT WAS FOR	B / N / Y	#	DECISION, AND WHO IT WAS FOR	B / N / Y
1			16		
2			17		
3			18		
4			19		
5			20		
6			21		
7			22		
8			23		
9			24		
10			25		
11			26		
12			27		
13			28		
14			29		
15			30		

The pattern, written as a strategy somebody handed you

Read off Meridian’s four years: “We serve mid-size logistics operators, plus whichever large enterprise asked us most recently, and we’re running a custom integrations business on the side.” Nobody chose it.

From module 5, They Asked for Your Strategy. You Opened the Roadmap.: youtu.be/uerj3F4aoys

Your Biggest Customer Just Asked for the Thing You Said No To.

How do you turn a strategy into a plan that doesn't collapse the first time a big customer calls?

Three weeks after the decline is written, the biggest customer asks for exactly the declined thing, with the renewal mentioned near the bottom of the email. Scored by the book with RICE, the request comes out on top, and MoSCoW's won't-have only means not this release. Neither framework can refuse anything.

A roadmap that survives the call promises outcomes owed to the customer you chose, and stops naming features. The decline gets an escalation rule, written the same day. And the customer still gets a real answer, from a phone call: what you're for, what the request would cost, and what would change your answer.

WATCH MODULE 6 youtu.be/IYDnb1NCtVY

THE PAGES FOR THIS MODULE

RICE, scored twice	27
Outcomes you owe	28
The escalation rule	29
When the request comes in: one phone call	30
What did the decline buy?	31

RICE, scored twice

RICE, a popular framework for ranking requests, scores them on reach, impact, confidence and effort, with the first three multiplied together and divided by the fourth. Score a big customer’s request by the book and it lands on top. Score the bet your strategy rests on the same way and the confidence term sinks it, because a strategy is a bet nobody can prove in advance.

HOW TO USE IT

- 1. Score the request in the column headed *The request*, filling in each term the way the framework describes it.
- 2. Score the bet your strategy rests on under *Your strategy’s bet*, on the same scales.
- 3. Compare the confidence lines. A named request in writing reads as high confidence; an untested insight reads as low, fifty percent, the bottom of the framework’s scale.

TERM	WHAT IT MEASURES	THE REQUEST	YOUR STRATEGY’S BET
Reach	How many customers this touches in a given period.		
Impact	How much it matters to them, on whatever scale you pick.		
Confidence	How good your evidence is: 100% high, 80% medium, 50% low. Anything under fifty is a shot in the dark.		
Effort	How much work it is. It’s the divisor, so low effort raises the score.		
Score	$\text{Reach} \times \text{impact} \times \text{confidence} \div \text{effort}$.		

MoSCoW, the other framework you’ll meet: must have, should have, could have, won’t have. The won’t-have bucket looks like a decline, but it’s scoped to the release you’re planning, and in some versions the W stands for wish. The closest either framework gets to a decline is a maybe with a date on it.

The reading that the confidence term favors a named request is the course’s own analysis; check it against the formula on this page. The explainer the module scored from was made by a company that sells a prioritization tool.

Outcomes you owe

A roadmap of features can always be outranked by a request for a better feature, so the only question it can answer is which yes comes first. A roadmap that promises outcomes owed to the customer you chose gives a request nothing to outbid.

HOW TO USE IT

- 1. Write the outcomes you owe the customers in your focus, each with a date.
- 2. Phrase each as a promise you could be held to. Meridian's: *by the end of the second quarter, a new customer with no IT department is running on their own inside a day.*
- 3. When a request arrives, ask whether it serves an outcome on this page, and answer the customer in those terms.

OUTCOME WE OWE, AND TO WHOM	BY WHEN	WORK NOW PLANNED TOWARD IT

The request that just arrived. Does it serve an outcome above?

A roadmap says what you intend to pursue. It says nothing about whether the problem is real, or whether your biggest customer is the only one who has it. That's module 7.

The escalation rule

A decline can be undone without anybody ever overruling it: the founder says yes to an account executive in a hallway, and by Thursday it's back on the roadmap. The escalation rule protects it. Write it at the same time you write the decline, because afterward everybody is too invested.

HOW TO USE IT

1. Copy the decline from your one-page strategy (module 5).
2. Write the three lines. The third does the work: if the answer is a number, say the number; if it's evidence, say what evidence; if nothing could change it, say that.
3. If you only do one thing this week, fill in line 1: who can say no without asking you. If you can't do that in one sentence, the decline isn't finished.

The decline

1. Who can say no on their own

2. What size of exception has to come to the founder

3. What has to be true for the answer to change

Meridian's rule. Any single customer request, the head of product decides. Anything that needs a new kind of engineering commitment goes to the founder. And the decline changes only if three customers in the chosen focus ask for the same thing.

The three-line rule is the course's own. If every no has to come from you, what you've written is a job description for yourself. No study compares companies with written escalation rules against companies without them; what the module stands behind is that a decline you wrote down can be pointed at, and one in your head gets relitigated every time.

When the request comes in: one phone call

Marty Cagan's high-integrity commitment says to give a date only after you understand what the customer needs and know what delivering it takes. A forty-person company can't build a prototype for every request, but it can afford the other half of his rule, *never accept proxies*: neither the salesperson nor the customer's own summary tells you what they need. Somebody from your side talks to them, and at forty people that's probably you.

HOW TO USE IT

- 1. Make the call even when the answer is going to be no. What somebody asks for and what they need aren't always the same thing; write both in the boxes below.
- 2. On the call, don't read them your strategy. Tell them what you're for, then the truth about what the request would cost, then what would change your answer.
- 3. Leave with a condition you can both check, so the customer's renewal hangs on something you can both go and look at.

WHAT THEY ASKED FOR	WHAT THEY MIGHT NEED

WHAT TO SAY ON THE CALL

1. What we're for

2. What the request would cost us

3. What would change our answer

At Meridian, the customer asked for a custom integration into their warehouse system because their operations team was retyping numbers from one screen into another, about forty minutes a day. What they might need is their data arriving somewhere once a day without anyone retyping it.

The module's sample wording for the third line: "If the operators we're built for need this too, we'll build it properly, for all of them. Give me time to find out whether you're the only one."

Cagan's rule is paraphrased in the module, and his firm sells transformation services; nothing the module found measures whether this works better.

From module 6, Your Biggest Customer Asked for the Thing You Said No To: youtu.be/IYDnb1NCtVY

What did the decline buy?

Everything in an escalation rule argues for no, so check the other direction too. A decline is supposed to buy something: the thing you're for gets better, and faster. And sometimes the answer should be yes, because a strategy is a bet and bets get updated.

HOW TO USE IT

1. Ask both questions of each decline on your one-page strategy.
2. If you can't answer the second, the decline was just expensive.
3. If a decline has turned into a yes, mark where the yes came from.

1 What did this decline cost me?

2 What did I buy with it?

IF THE ANSWER BECAME YES, WHERE DID THE YES COME FROM?

- ☐ The rule we wrote. For example, the third customer in our focus asking for the same thing, which is the market saying the insight was incomplete.
- ☐ Whoever was loudest that week.

A yes that came from the rule you wrote is one you can defend. A yes that came from whoever was loudest is one you'll repeat.

Your Biggest Customer Wants It. Does Anyone Else?



What evidence justifies committing engineering months?

A plus one on the request list feels like evidence. But a feature request describes something for you to build, so every plus one counts somebody proposing the same solution, and the list has no column for how many customers have the problem. Of Marty Cagan’s four big risks, value goes first, and for one customer’s request it becomes a plain question: does anyone besides them have this problem badly enough to do something about it?

Module 7 sizes discovery for a company without a product team: write the bar before the first call, ask about the past instead of the future, and keep one standing customer conversation a week. Meridian’s count came back two of sixty against a bar of three, and the decline held.

WATCH MODULE 7 youtu.be/fKcaAphdQQ0

THE PAGES FOR THIS MODULE

Supply and demand	33
The four big risks	34
Set the bar first	35
Ask about the past	36
A pace a forty-person company can keep	37

Supply and demand

Ryan Singer and Chris Spiek split the world into two sides. Supply is what you build and ship. Demand is what's happening in your customer's life, which you don't control. A feature request is written on the supply side, so the job is to find the demand underneath it.

HOW TO USE IT

- 1. List the requests with the most plus ones, or the one from your biggest customer.
- 2. For each, find out what's happening in the customer's week that produced it. The questions on this module's *Ask about the past* page are built for that.
- 3. Mark whether they're already doing something about it. That's the column the request list never had.

THE REQUEST (SUPPLY)	WHO ASKED	WHAT'S HAPPENING IN THEIR WEEK (DEMAND)	DOING SOMETHING TODAY?
<i>A custom integration into our warehouse system (Meridian's example)</i>	<i>The biggest customer</i>	<i>Someone retypes numbers from one screen into another, about forty minutes a day</i>	<i>Yes</i>

Singer and Spiek's own example: a request for permissions turned out to be about trust, after a contractor archived a shared project and it vanished for everyone else. Their alternative to the permissions feature was a warning on the archive screen. From their 2017 video for Demand Thinking, paraphrased in the module.

From module 7, Your Biggest Customer Wants It. Does Anyone Else?: youtu.be/fKcaAphdQQ0

The four big risks

Marty Cagan names four risks to deal with before you build anything, and advises tackling value and business viability early. Value goes first: if nobody else has the problem, you never have to answer the other three.

HOW TO USE IT

- 1. Write the request at the top.
- 2. Work down the risks in order, and stop at value if the answer is no.

The request

RISK	CAGAN'S QUESTION	FOR THIS REQUEST
Value	Will customers buy it, or users choose to use it?	
Usability	Can they figure out how?	
Feasibility	Can your engineers build it with the time, skills and technology you have?	
Business viability	Does it also work for the rest of your business?	

For one customer's request, value risk turns into a plain question: **does anyone besides them have this problem badly enough to do something about it?**

Cagan gives value and viability to the product manager; at Meridian, that's the head of product. His roles assume a product manager, a designer and a lead engineer, and adapting them to a company without that trio is the course's. Cagan's firm sells transformation services.

From module 7, Your Biggest Customer Wants It. Does Anyone Else?: youtu.be/fKcaAphdQQ0

Set the bar first

Before you talk to a single customer, write down how much evidence it will take to change the answer. Once you start looking, every answer starts to sound like agreement, and a maybe starts to sound like a yes. Set the bar at a number only real demand can reach, because polite interest will fill any bar you leave loose.

HOW TO USE IT

- 1. Pick the line on your request list with a single name next to it.
- 2. Before you call anybody, write the bar: who has to have the problem, how many of them, and what counts as having it.
- 3. Size the test with Cagan’s rule: a test should take an order of magnitude less time and effort than the build. A three-month build gets about a week and a half.
- 4. Count, then read the count against the bar you wrote first.

The request (the line with a single name next to it)

THE BAR, WRITTEN BEFORE THE FIRST CALL	
Who has to have the problem	
How many of them	
What counts as having it: something they’re already doing about it today	

BUILD ESTIMATE	TEST BUDGET (A TENTH)	THE COUNT	OVER THE BAR?
			☐ Yes ☐ No

Meridian’s bar: three customers in the focus who are doing something about the problem today. After two weeks of calls, nine said an integration would be nice and none of the nine was doing anything about it; two of sixty had the problem. Two is under the bar, and the decline held.

Setting the bar first is the course’s own practice; none of the sources it cites says to, and nobody has measured whether it leads to better decisions. A bar written before the count can be checked against the count, and that’s all the module claims for it. Building the request anyway is allowed, as long as you call it a bet. Cagan’s firm sells transformation services.

From module 7, Your Biggest Customer Wants It. Does Anyone Else?: youtu.be/fKcaAphdQQ0

Ask about the past

The questions you ask can wreck the answer. Ask someone whether they'd use a feature and they'll say yes, because nobody wants to be rude. Ask about the past instead: if somebody has the problem, they've already done something about it.

HOW TO USE IT

1. Before the call, check yourself against the three mistakes.
2. Use one or two of these as the last lines of a call you were already having.
3. Throw out compliments. There's nothing specific in them to use.
4. Listen for the budget: a spreadsheet somebody maintains, a person hired to deal with it, forty minutes a day of retyping. Nobody funds a workaround for a problem they don't have.

THREE MISTAKES, FROM THE MOM TEST

- ☐ You pitch your idea.
- ☐ You ask about hypotheticals.
- ☐ You talk too much.

ASK INSTEAD

- 1 Tell me about the last time that you encountered this problem.

- 2 What, if anything, have you done to try to solve this problem?

- 3 What are you doing about this today?

- 4 What was happening the day before you wrote to us?

The three mistakes are Eric Migicovsky's list, from Y Combinator's talk *How to Talk to Users*, built on the book *The Mom Test*. Questions 1 and 2 are quoted from Y Combinator's written recap of that talk. Question 3 is the course's one-question version. Question 4 is Ryan Singer and Chris Spiek's: ask when, not why, and the conversation moves out of the product and into the customer's week.



A pace a forty-person company can keep

The discovery tempo Teresa Torres and Marty Cagan describe assumes a product manager, a designer and an engineer doing discovery together every week, a team Meridian doesn't have. Torres, who coaches product teams, says continuous discovery can start with one interview a week. This log keeps that pace for a quarter.

HOW TO USE IT

- 1. Hold one standing conversation a week with a customer.
- 2. Borrow the people who already talk to customers. Support and account managers are on the phone all day, and Torres suggests using them to line up conversations.
- 3. End calls you were already having with: *What are you doing about this today?*
- 4. When a request would cost engineering months, give it a test measured in days, with the bar written first (the *Set the bar first* page).

WEEK	CUSTOMER	LINED UP BY	WHAT THEY'RE DOING ABOUT IT TODAY
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			

The assembled pace is the course's own; no source it found sizes discovery for a company this small.

From module 7, Your Biggest Customer Wants It. Does Anyone Else?: youtu.be/fKcaAphdQQ0

You Shipped It on Time. Nobody Asked If It Worked.



How do you know whether what you shipped did what it was supposed to?

A card in the Done column records that the work shipped. Whether it worked is a question the column can't answer. Numbers chosen after the results come in can all be true and still be bound to look good. Medical journals met the same problem by requiring trials to register what they'd measure before the first patient joined.

Module 8's scoreboard does that job for a product bet: six lines, written before the build. The module expects a scoreboard that tells the truth to be mostly misses, so it only works if a miss is safe to report. Meridian's setup flow reached three of six against a bar of four, and the head of product reported it as a miss.

WATCH MODULE 8 youtu.be/cwkZxcBrago

THE PAGES FOR THIS MODULE

The scoreboard, written before anyone knows the score	39
O'Reilly's hypothesis template	40
The column after Done	41
Reporting the miss	42

The scoreboard, written before anyone knows the score

One page, attached to the card before anybody builds anything. It fixes in advance what would count as having worked, so the result gets read against it. A measure picked after the result is in is bound to look good.

HOW TO USE IT

1. Start with the outcome: what a customer will do differently once the work exists. Josh Seiden defines an outcome as a change in human behavior that drives business results.
2. Count what customers do. A customer can call the setup easy on Tuesday and still call support on Wednesday, and the call is what you count.
3. Write today's number, then the bar: the number that would mean it worked. Without today's number, you're judging the result against a guess.
4. Put a date and a name on it. Without a date, the check slides into next quarter; with a name, reporting the result is that person's job, whatever it turns out to be.

LINE	WHAT GOES HERE	THIS CARD
1. The outcome	What a customer will do differently once the work exists	
2. How you'll count it	The behavior you'll count, and where the count comes from	
3. Today's number	The count before the work ships	
4. The bar	The number that would mean it worked	
5. The date you'll check	When you read the count against the bar	
6. The name on the result	Who reports it, hit or miss	

Meridian's scoreboard. Outcome: a new customer publishes a first schedule within a day of signing up, without calling support. Today's number: one of last quarter's six. Bar: four of the next six. Date: the end of the quarter. Name: the head of product. Result: three. A miss, though triple the quarter before.

The course assembled this page from the sources it quotes. Trial registration shows that writing it first catches a switched measure; whether it makes a product team decide better is untested. For the date, John Cutler suggests a promise small enough to keep, like revisiting each decision a month or two after you make it.

From module 8, You Shipped It on Time. Nobody Asked If It Worked.: youtu.be/cwkZxcBrago



O'Reilly's hypothesis template

Barry O'Reilly published this in 2013 as a way of writing down a bet. Its last sentence does what a trial's registration does: it names, before anyone builds anything, what would count as having worked.

HOW TO USE IT

- 1. Fill in both sentences before the work starts.
- 2. Test the last line. If it only says customers will like it, it can be read as a yes whatever happens; name something you could count.
- 3. Carry the signal onto the scoreboard, which adds what the template doesn't ask for: a number to beat, a date to check it, and a person to report it.

We believe this capability ...

will result in this outcome ...

We will have confidence to proceed when we see a measurable signal ...

A REAL LAST LINE	A LAST LINE THAT CAN'T FAIL
Names what would count as having worked	"Customers will like it" reads as a yes whatever happens

Template wording quoted from O'Reilly. The comparison with clinical trial registration is the course's analogy, not his claim.

The column after Done

John Cutler’s fix for shipped work nobody checks starts on the board itself: a column after Done, where the team checks what the work did. It gives the question a place on the board. A check made only after the fact can still pick whichever numbers look good, which is why the scoreboard comes first.

HOW TO USE IT

- 1. List what’s sitting in Done from the last quarter.
- 2. For each card, mark whether anyone checked what it did, and what they found.
- 3. Any card with nothing in the last column is a candidate for a scoreboard next time.
- 4. Check either of Cutler’s two signs below if it describes your team.

DONE: WHAT SHIPPED	CHECKED?	AFTER DONE: WHAT IT DID
	☐ Yes ☐ No	
	☐ Yes ☐ No	
	☐ Yes ☐ No	
	☐ Yes ☐ No	
	☐ Yes ☐ No	
	☐ Yes ☐ No	
	☐ Yes ☐ No	

TWO OF CUTLER’S 12 SIGNS YOU’RE WORKING IN A FEATURE FACTORY

- ☐ You have no idea if your work worked. (His first sign: no measurement.)
- ☐ Developers have passing tests, but product managers do not. A developer writes the test before the code ships; a product decision can ship with nothing like that.

Cutler’s list is from 2016 and the column comes from his 2019 talk, paraphrased in the module. A feature factory, in his term, keeps shipping features without checking what any of them did. The module uses two of his twelve signs, and only those two are here.

From module 8, You Shipped It on Time. Nobody Asked If It Worked.: youtu.be/cwkZxcBrago

Reporting the miss

The scoreboard only works if a miss is safe to report. Everyone in the room watches what happens to the first person who reports one, and takes their cue from it. Use this page at the review where a result comes in.

HOW TO USE IT

1. Read the bar first, then the result, then call it a hit or a miss, even when the result improved.
2. Write what the miss taught you and where it goes next.
3. If the miss shows the measure itself was wrong, change it on the next card's scoreboard and say why. The result in hand still counts against the bar it was written with.
4. The last question should come back no. If something happened to the person who reported the miss, expect the details to stop coming.

THE BAR	THE RESULT	HIT OR MISS

1 What did the miss teach us?

2 What goes on a new card, with a scoreboard of its own?

3 Was the measure itself wrong? If so, what does the next scoreboard say instead, and why?

4 Did anything happen to the person who reported it?

☐ Yes ☐ No

Meridian's review. Bar four of six, result three: a miss, though three was triple the quarter before. All three customers who called got stuck at the same step, setting up overtime rules, so templates for the common overtime rules went on a new card. The engineers heard their work called a miss in a meeting where nothing happened to anyone for it.

John Allspaw's no-blame rule is about outages at Etsy: someone who can describe exactly what happened without being punished teaches everyone how the site came down. Applying it to a product bet that missed is the course's own extension. A perfect scoreboard can't tell you whether a team misses less or reports less.

From module 8, You Shipped It on Time. Nobody Asked If It Worked.: youtu.be/cwkZxcBrago



You Hired Someone Good. Then You Answered for Them.



What is the leadership work that isn't about the product at all?

When a head of product answers the hard question for a new hire in a review, the room learns the decision is still hers. The module found that the most-watched tools on product skills measure the person, and none of them says which decisions to hand a new hire, or how much of each.

Medicine had the same problem with new doctors and solved it by granting trust one activity at a time, on five levels of supervision. Module 9 rewrites those levels for product work and puts them on one page per new hire: the four decisions down the left, what they'd have to show you, the level they're at today, a weekly one-on-one and a date. The coaching changes with the level: show, then ask, then leave the room.

WATCH MODULE 9 youtu.be/dVwygFKc5dU

THE PAGES FOR THIS MODULE

One page for a new hire	44
Five levels, and how to coach at each	45
The weekly one-on-one	46
Gap analysis	47

One page for a new hire

Whether a new hire is ready has no answer until you say ready for what, and with how much supervision. One page per hire answers it decision by decision. The levels come from medicine; using them on product decisions is the course’s own idea.

HOW TO USE IT

- 1. Clip your one-page strategy (module 5) to the top. Your new hire reads it before anything else.
- 2. For each decision you hand over, write what they’d have to show you: something you could watch them do and then judge.
- 3. Mark the level they’re at today, one to five. The levels are on the next page.
- 4. Set a weekly one-on-one, and a date a few months out when the decisions you’ve handed over should be at level four.

DECISION	HANDED OVER?	WHAT THEY'D HAVE TO SHOW YOU	LEVEL TODAY (1-5)
Which problems we solve, and for whom	☐ Yes ☐ No		
What evidence justifies solving them	☐ Yes ☐ No		
What we decline	☐ Yes ☐ No		
What outcome we expect, and how we'd know	☐ Yes ☐ No		

WEEKLY ONE-ON-ONE (DAY AND TIME)	LEVEL FOUR BY (DATE)

For the evidence decision, the module’s example: before a customer call, they write down how many customers would need to have the problem. After it, they can say what the customer has done about it, kept apart from what the customer promised.

Nobody has tested a page like this on product teams. If the date comes and the hire isn’t at level four, the next step is a different role for the hire. In an emergency you’ll take a decision back; say so out loud, and say when it goes back.

From module 9, You Hired Someone Good. Then You Answered for Them.: youtu.be/dVwygFKc5dU

Five levels, and how to coach at each

Medicine grants trust to new doctors one activity at a time, on a scale of five levels of supervision. The course rewrites the levels for product work, and the coaching changes with the level.

HOW TO USE IT

1. Use the product column to mark each decision's level on a new hire's page.
2. Coach to the level the decision is at: show at one and two, ask at three, leave the room at four.
3. When you explain something, ask how they'd apply it. The coaching video the module cites points out that "Does that make sense?" gets a yes almost every time.

LEVEL	IN MEDICINE	FOR PRODUCT WORK	HOW YOU COACH
1	The trainee only watches	They watch you do it	Show them
2	They do it with the supervisor right there	They do it with you there	Show them
3	They do it with the supervisor nearby, ready when called	They do it, and you check it before it goes out	Ask the questions
4	They do it unsupervised	They do it alone, and tell you when it misses	Leave the room. The hard question in a review is theirs to answer.
5	They supervise someone else doing it	They teach it	Not covered in the module

Alongside competence. A 2017 paper for supervisors, led by Carrie Chen, lists what has to sit alongside skill: integrity, reliability, and humility, the willingness to ask for help if needed. A supervising doctor weighs whether a resident will call for help. You're weighing whether your new hire will tell you the bar was missed before somebody else finds it.

Granting trust one activity at a time comes from Olle ten Cate, a Dutch medical educator (2005); the five levels are the scale medicine settled on, paraphrased here. The course reads coaching by showing and coaching by asking as both right, at different levels.



The weekly one-on-one

Marty Cagan’s written rule is a one-on-one of at least thirty minutes every week, never canceled. For a new product manager who isn’t competent yet, he says it may be two or three times a week, or even daily. In his words, its primary purpose is to help the product manager develop and improve. The page for the new hire sets the agenda.

HOW TO USE IT

- 1. Each time you meet, pick one decision from the page and work on moving it up one level.
- 2. When it moves, say so to everyone who sends that kind of work.
- 3. Log each meeting here.

DATE	DECISION WORKED ON	LEVEL: FROM, TO	TOLD (WHO SENDS THAT KIND OF WORK)

Competence, or potential. Cagan’s rule is to hire for competence, or to hire on potential only if the person doing the hiring is willing and able to coach the new hire to competence. Writing the page first shows you how much coaching a candidate would need. Meridian’s head of product hired the support lead on potential, knowing the cost was months of her own coaching, and the two of them met three times a week at first.

Cagan’s firm sells coaching and transformation work.

From module 9, You Hired Someone Good. Then You Answered for Them.: youtu.be/dVwygFKc5dU

Gap analysis

Marty Cagan’s tool for deciding where to spend your coaching: rate how important each skill is for the team, rate your product manager against it, and coach the biggest gaps first. It rates skills, and the page for a new hire rates decisions.

HOW TO USE IT

- 1. List the skills this team’s product work needs.
- 2. Rate how important each one is, from one to ten.
- 3. Rate your product manager against each one, from one to ten.
- 4. Subtract where they are from how important it is, and coach the biggest gaps first. Use the result beside the page for a new hire: the gaps tell you where to coach, and the page tells you which decisions to hand over.

SKILL	HOW IMPORTANT (1-10)	WHERE THEY ARE (1-10)	GAP

From Cagan’s published coaching tools (SVPG, 2019). His firm sells coaching and transformation work.

From module 9, You Hired Someone Good. Then You Answered for Them.: youtu.be/dVwygFKc5dU

Your Head of Sales Blamed Product. Nobody Showed Him Why.



How does product leadership work across the other executives?

Every decision product makes lands on somebody else’s desk. A decision that arrives without a reason looks like product getting in the way, and people find a way around it.

Module 10 asks what each of the other executives needs from product to do their own job, and gives it to them before they need it, on one sheet with four columns: sales, the founder, engineering and the board. Most of it comes off pages you already have, the strategy and the scoreboard. At Meridian, a month after the board meeting, the head of sales explained a decline to a customer himself, with the reason and what would change it.

WATCH MODULE 10 youtu.be/nLC7ZpOZQvs

THE PAGES FOR THIS MODULE

The sheet	49
When the CEO wants an exception	50
No surprises: the board	51

The sheet

One sheet, four columns, one for each person a product decision lands on. Each column is a conversation you keep having with one of them.

HOW TO USE IT

1. Fill each column, mostly from your strategy page (module 5) and your scoreboard (module 8).
2. Walk each person through their column before they need it. At Meridian, the head of product walked the head of sales through the sales column before the next Monday meeting, with the strategy page.
3. This week: pick one thing you've said no to, and walk your head of sales through the reason and what would change it.

Sales

What it can promise this quarter. What product has said no to, and why. What would change that answer.

The founder

The options, and what each one would delay, talked through with the whole leadership team.

Engineering

Which parts of the product are settled and which are bets. A settled piece gets built to last; a bet gets built so it's cheap to change.

The board

The scoreboard: the bar, written before the work started, and the result next to it. Any miss told to each board member before the meeting.

Meridian's head of sales, to a prospect: "We don't build custom warehouse integrations, because our customers need to be running on the first day. If three operators like you need the same thing, we'll look again."

The four-column layout is the course's own. No study the module found compares companies holding these four conversations with companies that don't. Module 11 puts all four conversations on a calendar.

When the CEO wants an exception

Whenever the CEO wants something that breaks the plan, lay out the options and what each would delay, and talk them through with the CEO's other direct reports in the room. The whole leadership team then picks which cost to pay, in the open, and nobody has to watch product and the CEO argue it out.

HOW TO USE IT

- 1. Write down what the CEO wants.
- 2. List the options, with what each one would delay.
- 3. Bring them to the leadership meeting, and record what the team picked.

What the CEO wants

OPTION	WHAT IT WOULD DELAY

Who was in the room

What the team picked, and the cost it chose to pay

Meridian's two options. Build the integration for the big prospect, and the overtime templates slip about three months. Keep the decline, let the prospect go, and the head of sales works the operators inside the focus. The founder kept the decline.

The founder could still choose the integration; it's his company, and that's a legitimate call. If he does, he makes it in front of the team, knowing the cost. The approach is from a 2019 episode of Fearless Product, paraphrased in the module. Rich Mironov, who coaches product executives, answers such requests with a question about work already promised: these roadmap items help your group, so do you want to push them back for this?

From module 10, Your Head of Sales Blamed Product: youtu.be/nLC7ZpOZQvs

No surprises: the board

Every quarter of product work spends runway, so the board needs to know what it bought. The investors the module reads keep coming back to one rule, which Mark Suster calls the golden rule of board meetings: no surprises.

HOW TO USE IT

- 1. Work down the checklist before each board meeting.
- 2. Fill in the bar, the result and what the miss taught you. That's your report, misses included: module 8 expects a scoreboard that tells the truth to be mostly misses, and the board meeting is where you report them.

STEP	WHEN (MERIDIAN'S TIMING)	DONE
Call each board member about any miss, so nobody hears bad news for the first time in the room (Fred Wilson).	A week before	☐
Send the materials early, so the meeting can be a discussion (Brad Feld).	Three days before	☐
Report what you've done against what you said you'd do (Brett Fox).	In the meeting	☐
Bring the scoreboard: the bar, written before the work started, and the result next to it.	In the meeting	☐

What we said we'd do (the bar)

What we did (the result)

What the miss taught us

Meridian's call to each board member: "Before Thursday, I want you to hear this from me. We set a bar of four, we got three, and here's what we learned."

Suster, Wilson, Fox and Feld write for CEOs; applying their advice to a head of product is the course's own step. It's investor and practitioner advice, not research.

You're Planning Next Quarter. You Never Checked This One.



What calendar makes all of this repeatable instead of heroic?

A date written on a page doesn't remind anybody of anything. The scoreboard's check date passes unread, and a decline with no date for its next look comes back whenever something breaks, raised by the person it hurt.

Module 11 puts every decision on one page with the thirteen weeks of the quarter across the top, each looked at again on its own schedule: the evidence weekly, the outcome on its scoreboard's date, and the problem and decline decisions once a quarter, at a strategy refresh. In between, pressure waits for the date, and evidence that meets the condition written beside a decision can move the date up.

WATCH MODULE 11 youtu.be/T7TnIAWd4Gw

THE PAGES FOR THIS MODULE

The quarter on one page	53
Holding a decision until its date	54

The quarter on one page

Every decision you own can have a date when it gets looked at again, and all of those dates fit on one page. Each decision changes at its own speed, so each gets its own schedule. Give a decision a date, and you can hold it until then.

HOW TO USE IT

1. Start with the decline you've defended the most times. Write what would change it, and put the date it gets looked at again on the calendar everyone uses.
2. Mark each row's dates in the weeks they fall.
3. Put every date on the shared calendar. The Monday meeting looks at the coming week first. The strategy refresh is the one new meeting; the end-of-quarter review is separate, and Meridian already held one for its scoreboards.
4. At the end of the quarter, hold the review before anybody plans the next one.

WEEK	1	2	3	4	5	6	7	8	9	10	11	12	13
Which problems: once a quarter, at the strategy refresh													
What evidence: every week, from customer calls													
What we decline: at the refresh, with what would change it													
What outcome: on its scoreboard's date													
Product manager's one-on-one: every week													
Monday meeting, product's standing item: every week													
Board calls: the week before each board meeting													
The review: end of quarter, before the next plan													

The layout is the course's own, and quarterly is its choice. Marty Cagan, who sells product training through his firm, writes that a product strategy changes at least yearly if not quarterly; that range is his judgment, and nobody the module found has measured it.

From module 11, You're Planning Next Quarter. You Never Checked This One.: youtu.be/T7TnIAWd4Gw

Holding a decision until its date

A no with a date on it can wait out a bad week. Every Monday, product brings one standing item: what did we say no to this week, why, and when will we look at it again? When something turns up between dates, one test decides whether it waits.

HOW TO USE IT

- 1. Log product's standing item every Monday.
- 2. When something turns up mid-quarter, ask whether it touches the condition written beside the decision. A lost deal from a customer outside the market you chose is pressure: somebody wants the decision changed. Another customer inside that market, with the same workaround, is evidence: it meets the condition.
- 3. Pressure waits for the date. Evidence can move the date up.

WHAT WE SAID NO TO THIS WEEK	WHY	LOOKED AT AGAIN ON

PRESSURE OR EVIDENCE?

WHAT TURNED UP MID-QUARTER	TOUCHES THE CONDITION?	WAITS, OR MOVES THE DATE TO
	☐ Yes ☐ No	
	☐ Yes ☐ No	
	☐ Yes ☐ No	

Meridian's quarter. In week five the founder forwarded a prospect's email from outside the focus, and it waited for the refresh set for week thirteen. In week eight a third operator inside the focus turned up with the same need, and the head of product moved the refresh up to week nine.

Roger Martin checks a strategy by writing down what would have to be true for each choice and continuing to check; when something it rests on stops being true, "that is precisely the time to get to work." Dan Martell, who coaches software founders, parks a mid-quarter idea in the planning document, so the person is heard and the idea has a date. Treating pressure and evidence differently is the course's way of reconciling Martin with that parking rule; nobody has tested whether parking ideas reduces relitigating.

From module 11, You're Planning Next Quarter. You Never Checked This One.: youtu.be/T7TnIAWd4Gw

Your Board Wants a Chief Product Officer. Which Decisions Would You Hand Over?

How do you know the founder has stopped being the right person to hold product, and what exactly do you hire then?

Ask when to hand product to someone else, and every rule you find is a number: a million dollars in revenue, four or five million, ten to fifteen engineers. None of them says what the new person would decide. Module 12 brings the question back to the four decisions and asks whether each one is held on purpose.

Three yeses mark a decision held on purpose: a date that's kept, changes brought to the person who holds it, and a written condition that changes it. For the founder's own decisions there's a fourth: would you hire yourself today? A decision that gets a no moves to someone else, and whether it's an executive or an execution decision tells you who should take it. Meridian's page came back with every yes, so its founder told the board it isn't hiring a chief product officer yet.

WATCH MODULE 12 youtu.be/-xP8mbp6mK0

THE PAGES FOR THIS MODULE

Held on purpose, or by default?	56
Who takes over a decision	57
The handover	58

Held on purpose, or by default?

Whether to hand over product comes down to the four decisions, and for each one, whether it's held on purpose. This is module 2's audit filled in again, with the yeses beside each name. Three yeses (and, for your own decisions, a fourth), and the decision is held on purpose. Anything less, and it's held by default, whatever title the person holding it has.

HOW TO USE IT

- 1. Beside each decision, write the name of whoever makes that call today.
- 2. Ask the three yeses. Does it have a date when it gets looked at again, on the calendar everyone reads, and does the date get kept? Does anyone who wants it changed bring it to the person who holds it? Is a condition written beside it, and does the decision change when the condition is met?
- 3. For each decision with your own name beside it, ask a fourth, every six months or so: if you were hiring someone to make this decision today, at the size the company is now, would you hire yourself?
- 4. Circle each decision that gets its yeses. The rest are the ones to move; the next page says to whom.

DECISION	ALTITUDE	HELD BY	DATE KEPT	BROUGHT TO HOLDER	CONDITION ACTS	HIRE YOURSELF?
Which problems we solve, and for whom	Executive		☐	☐	☐	☐
What we decline	Executive		☐	☐	☐	☐
What evidence justifies solving them	Execution		☐	☐	☐	☐
What outcome we expect, and how we'd know	Execution		☐	☐	☐	☐

Meridian's page. The founder holds which problems and the declines, at a quarterly strategy refresh with his leadership team in the room. The product manager holds the evidence and the outcome. The head of product holds none of the four herself: she keeps the calendar, coaches the product manager, and walks sales, engineering and the board through each decision. Every decision got its yeses, and at forty people the founder would still hire himself for his two.

The three yeses and the fourth question are the course's own test, and there's no study behind them. Keeping product is a decision too, and it gets a date like any other: Meridian's founder will fill in this page again at the next strategy refresh.

Who takes over a decision

Keep every decision that gets its yeses on the previous page. Anything that gets a no moves to someone else, and whether it's an executive or an execution decision tells you who should take it.

HOW TO USE IT

- 1. Find your page's result in the left column.
- 2. Before anyone's hired, write on your *Held on purpose* page which decisions move to the new person, and on what date.

WHAT THE PAGE SHOWS	WHAT TO DO
All four decisions get their yeses, including the fourth for your own.	Keep them. You don't need to hire anyone to run product yet, and you can tell your board so.
The execution decisions are held by default: nobody owns the evidence, or nobody reads the scoreboards.	Hire someone to do that work under your strategy: a product manager or a director.
The executive decisions are held by default, and one of two reasons from Marty Cagan, whose firm trains product leaders, applies: the CEO isn't strong at product, or the CEO's time and interests have gone to other things.	Those two decisions need a product leader who has the time.
Both pairs are held by default.	Start with the product leader, because whoever holds the direction decides what the execution work is.

Full-time or fractional. The product leader can be full-time, or a fractional chief product officer: somebody senior who works part-time for several companies at once. Dave Saunders works as a fractional chief product officer himself, and says so in the module; weigh what this page says about the option with that in mind. Tomoko Yokoi, who studies fractional executives at IMD, says to figure out the work that needs doing before you pick the role, and agreed that a lot of work may need the full-time leader. Amy Bonsall, who has worked as a fractional chief product officer, named the limit: you're not there every hour of every day. The research so far asks fractional executives themselves, and none of it compares their results with a full-time hire's.

Founder mode is Paul Graham's name for a founder who stays in the details and talks to people two and three levels down. Done well, it's module 3's third answer, keeping the decisions on purpose. Graham predicted that founders unable to delegate even what they should would use founder mode as their excuse; the page tells the two apart, because a decision held out of habit gets at least one no.

From module 12, Your Board Wants a Chief Product Officer: youtu.be/-xP8mbp6mK0



The handover

A decision only becomes its new owner's when the rest of the company treats it that way. This page moves each decision on a date and makes sure the people who used to bring it to you know where it went.

HOW TO USE IT

- 1. Fill in a row for each decision that moves, then work down the checklist.
- 2. Once the new leader starts, agree on the pair of promises at the bottom of the page.

DECISION	NEW OWNER	MOVES ON	TOLD
Which problems we solve, and for whom			
What evidence justifies solving them			
What we decline			
What outcome we expect, and how we'd know			

BEFORE AND AFTER THE MOVE

- ☐ The page says which decisions move, and on what date, before anyone's hired.
- ☐ Everyone who used to bring each decision to you has been told where it went.
- ☐ You sent the next request for each decision to its new owner yourself.
- ☐ You're staying in the room for the big decisions for the first few months.

A PAIR OF PROMISES

The product leader Brings every hard decision as a list of options.	The CEO Says when the list is wrong, without quietly adding an option of his own.
--	--

Staying in the room for the first few months is Insight Partners' advice, paraphrased in the module: a new leader's ownership of the strategy grows over time. The pair of promises is one panelist's fix, from an Andreessen Horowitz podcast panel in 2019, and both promises assume the founder has already decided which calls to hand over.



THE COURSE

Watch all *twelve modules*

The Operator's Guide to Product Leadership is free on YouTube, in twelve parts. Every page in this guide links to the module it comes from, and the whole series is in one playlist: youtube.com/playlist?list=PLSARBRJT8EDc.

More from Dave Saunders at baserealitygroup.com.