

Managing the Machine Team

How to Lead People and AI Without Losing Accountability

Ngo Trung Kien

Contents

Copyright and publication notice	1
Introduction: the promise and the boundary	3
Visual frontispiece	4
0.1 Who this book is for	4
0.2 Thesis and falsification condition	4
1 Stop Calling AI a Teammate Too Soon	7
1.1 In plain terms	7
1.1.1 Chapter contract	7
1.2 The decision at stake	7
1.3 Working model	8
1.3.1 See the chapter in practice	8
1.3.2 Chapter-specific operating protocol	10
1.3.3 Operating workflow	12
1.3.4 Evidence and control graph	13
1.4 Worked composite case file — illustrative, not empirical evidence	14
1.4.1 Scenario trace	15
1.4.2 Artifact handoff	16
1.4.3 Evidence application	16
1.4.4 Transfer test	16
1.5 Measurement and recovery decision	16
1.6 Practice lab	17
1.7 Limits and judgment	17
1.8 Takeaways	18
1.9 Reflection	18
2 Map Work Into Decisions and Production	19
2.1 In plain terms	19
2.1.1 Chapter contract	19
2.2 The decision at stake	19
2.3 Working model	20
2.3.1 See the chapter in practice	20
2.3.2 Chapter-specific operating protocol	22
2.3.3 Operating workflow	24
2.3.4 Evidence and control graph	25
2.4 Worked composite case file — illustrative, not empirical evidence	26
2.4.1 Scenario trace	27
2.4.2 Artifact handoff	28
2.4.3 Evidence application	28
2.4.4 Transfer test	28

2.5	Measurement and recovery decision	29
2.6	Practice lab	29
2.7	Limits and judgment	30
2.8	Takeaways	30
2.9	Reflection	30
3	Set Delegation Levels	31
3.1	In plain terms	31
3.1.1	Chapter contract	31
3.2	The decision at stake	31
3.3	Working model	32
3.3.1	See the chapter in practice	32
3.3.2	Chapter-specific operating protocol	34
3.3.3	Operating workflow	36
3.3.4	Evidence and control graph	37
3.4	Worked composite case file — illustrative, not empirical evidence	38
3.4.1	Scenario trace	39
3.4.2	Artifact handoff	40
3.4.3	Evidence application	40
3.4.4	Transfer test	40
3.5	Measurement and recovery decision	40
3.6	Practice lab	41
3.7	Limits and judgment	41
3.8	Takeaways	42
3.9	Reflection	42
4	Write the Decision Rights	43
4.1	In plain terms	43
4.1.1	Chapter contract	43
4.2	The decision at stake	43
4.3	Working model	44
4.3.1	See the chapter in practice	44
4.3.2	Chapter-specific operating protocol	46
4.3.3	Operating workflow	48
4.3.4	Evidence and control graph	49
4.4	Worked composite case file — illustrative, not empirical evidence	50
4.4.1	Scenario trace	51
4.4.2	Artifact handoff	52
4.4.3	Evidence application	52
4.4.4	Transfer test	52
4.5	Measurement and recovery decision	52
4.6	Practice lab	53
4.7	Limits and judgment	53
4.8	Takeaways	54
4.9	Reflection	54

5	Create Evidence Handoffs	55
5.1	In plain terms	55
5.1.1	Chapter contract	55
5.2	The decision at stake	55
5.3	Working model	56
5.3.1	See the chapter in practice	56
5.3.2	Chapter-specific operating protocol	58
5.3.3	Operating workflow	60
5.3.4	Evidence and control graph	61
5.4	Worked composite case file — illustrative, not empirical evidence	62
5.4.1	Scenario trace	63
5.4.2	Artifact handoff	64
5.4.3	Evidence application	64
5.4.4	Transfer test	64
5.5	Measurement and recovery decision	64
5.6	Practice lab	65
5.7	Limits and judgment	65
5.8	Takeaways	66
5.9	Reflection	66
6	Budget Human Review Capacity	67
6.1	In plain terms	67
6.1.1	Chapter contract	67
6.2	The decision at stake	67
6.3	Working model	68
6.3.1	See the chapter in practice	68
6.4	Review-capacity queue example	70
6.4.1	Chapter-specific operating protocol	70
6.4.2	Operating workflow	72
6.4.3	Evidence and control graph	73
6.5	Worked composite case file — illustrative, not empirical evidence	74
6.5.1	Scenario trace	75
6.5.2	Artifact handoff	76
6.5.3	Evidence application	76
6.5.4	Transfer test	76
6.6	Measurement and recovery decision	76
6.7	Practice lab	77
6.8	Limits and judgment	77
6.9	Takeaways	78
6.10	Reflection	78
	Midpoint visual synthesis	79
7	Lead Through Exceptions	81
7.1	In plain terms	81
7.1.1	Chapter contract	81
7.2	The decision at stake	81

7.3	Working model	82
7.3.1	See the chapter in practice	82
7.3.2	Chapter-specific operating protocol	84
7.3.3	Operating workflow	86
7.3.4	Evidence and control graph	87
7.4	Worked composite case file — illustrative, not empirical evidence	88
7.4.1	Scenario trace	89
7.4.2	Artifact handoff	90
7.4.3	Evidence application	90
7.4.4	Transfer test	90
7.5	Measurement and recovery decision	90
7.6	Practice lab	91
7.7	Limits and judgment	91
7.8	Takeaways	92
7.9	Reflection	92
8	Measure Team Performance as a System	93
8.1	In plain terms	93
8.1.1	Chapter contract	93
8.2	The decision at stake	93
8.3	Working model	94
8.3.1	See the chapter in practice	94
8.3.2	Chapter-specific operating protocol	96
8.3.3	Operating workflow	98
8.3.4	Evidence and control graph	99
8.4	Worked composite case file — illustrative, not empirical evidence	100
8.4.1	Scenario trace	101
8.4.2	Artifact handoff	102
8.4.3	Evidence application	102
8.4.4	Transfer test	102
8.5	Measurement and recovery decision	102
8.6	Practice lab	103
8.7	Limits and judgment	103
8.8	Takeaways	104
8.9	Reflection	104
9	Protect Voice and Psychological Safety	105
9.1	In plain terms	105
9.1.1	Chapter contract	105
9.2	The decision at stake	105
9.3	Working model	106
9.3.1	See the chapter in practice	106
9.3.2	Chapter-specific operating protocol	108
9.3.3	Operating workflow	110
9.3.4	Evidence and control graph	111
9.4	Worked composite case file — illustrative, not empirical evidence	112

9.4.1	Scenario trace	113
9.4.2	Artifact handoff	114
9.4.3	Evidence application	114
9.4.4	Transfer test	114
9.5	Measurement and recovery decision	115
9.6	Practice lab	115
9.7	Limits and judgment	116
9.8	Takeaways	116
9.9	Reflection	116
10	Manage Vendor and Model Change	117
10.1	In plain terms	117
10.1.1	Chapter contract	117
10.2	The decision at stake	117
10.3	Working model	118
10.3.1	See the chapter in practice	118
10.3.2	Chapter-specific operating protocol	120
10.3.3	Operating workflow	122
10.3.4	Evidence and control graph	123
10.4	Worked composite case file — illustrative, not empirical evidence	124
10.4.1	Scenario trace	125
10.4.2	Artifact handoff	126
10.4.3	Evidence application	126
10.4.4	Transfer test	126
10.5	Measurement and recovery decision	127
10.6	Practice lab	127
10.7	Limits and judgment	128
10.8	Takeaways	128
10.9	Reflection	128
11	Develop Human Capability Deliberately	129
11.1	In plain terms	129
11.1.1	Chapter contract	129
11.2	The decision at stake	129
11.3	Working model	130
11.3.1	See the chapter in practice	130
11.3.2	Chapter-specific operating protocol	132
11.3.3	Operating workflow	134
11.3.4	Evidence and control graph	135
11.4	Worked composite case file — illustrative, not empirical evidence	136
11.4.1	Scenario trace	137
11.4.2	Artifact handoff	138
11.4.3	Evidence application	138
11.4.4	Transfer test	138
11.5	Measurement and recovery decision	139
11.6	Practice lab	139

11.7 Limits and judgment 140

11.8 Takeaways 140

11.9 Reflection 140

12 Run the Management Cadence 141

12.1 In plain terms 141

12.1.1 Chapter contract 141

12.2 The decision at stake 141

12.3 Working model 142

12.3.1 See the chapter in practice 142

12.3.2 Chapter-specific operating protocol 144

12.3.3 Operating workflow 146

12.3.4 Evidence and control graph 147

12.4 Worked composite case file — illustrative, not empirical evidence148

12.4.1 Scenario trace 149

12.4.2 Artifact handoff 150

12.4.3 Evidence application 150

12.4.4 Transfer test 150

12.5 Measurement and recovery decision 150

12.6 Practice lab 151

12.7 Limits and judgment 151

12.8 Takeaways 152

12.9 Reflection 152

Closing: keep the evidence visible 153

A compact implementation workbook 155

Decision register 156

Evidence scorecard 157

Exception log 158

Ninety-day plan 159

Control and test matrix 160

Workflow map 161

13 References 163

Stop Calling AI a Teammate Too Soon

In plain terms

Metaphors shape expectations, so leaders must separate capability from agency and accountability. In a human–AI production team, the practical question is **What confusion follows from treating software as a colleague?** For managers responsible for teams that use generative AI, copilots, automation, or agents, the answer counts only when **Describe each AI system by inputs, outputs, permissions, limits, and accountable owner** produces reviewable evidence for **unowned decisions found** without normalizing **using friendly language to hide missing responsibility**.

Chapter contract

DECIDE

What confusion follows from treating software as a colleague?

BUILD

Describe each AI system by inputs, outputs, permissions, limits, and accountable owner.

MEASURE

unowned decisions found

STOP OR REDESIGN WHEN

using friendly language to hide missing responsibility

The decision at stake

For **Stop Calling AI a Teammate Too Soon**, the accountable manager must decide **what confusion follows from treating software as a colleague?** before the shared assignment crosses from discussion into consequence for employees, managers, and customers. The minimum decision record is the artifact **Describe each AI system by inputs, outputs, permissions, limits, and accountable owner**, the observable **unowned decisions found**, and the explicit hold condition **using friendly language to hide missing responsibility**. If any Chapter 1 field is missing, the safe default is to return the decision to a named person with sufficient time and context.

The bounded **Stop Calling AI a Teammate Too Soon** decision advances the book's promise to design roles, decisions, review, and learning so AI increases team capacity without obscuring responsibility. It does not endorse a product or predetermine the Chapter 1 answer; it creates a record that another person can challenge at each queue review and team retrospective.

Working model

Read the **Stop Calling AI a Teammate Too Soon** model as four linked fields. **Outcome:** Metaphors shape expectations, so leaders must separate capability from agency and accountability. **Boundary:** the design must make **using friendly language to hide missing responsibility** visible rather than convenient. **Evidence:** interpret **unowned decisions found** together with review effort and excluded cases. **Authority:** the accountable manager may continue, narrow, escalate, or stop the shared assignment. For **Stop Calling AI a Teammate Too Soon**, Reference [1] from National Institute of Standards and Technology, *AI Risk Management Framework*, informs a Chapter 1 boundary; it is not evidence that the local method or illustrative target will work.

See the chapter in practice

The **stop calling ai a teammate too soon in practice** maps *Managing the Machine Team* Chapter 1 into a field scene before its abstract models begin. In one frame, it holds the question **What confusion follows from treating software as a colleague?**, the artifact **Describe each AI system by inputs, outputs, permissions, limits, and accountable owner.**, the proof signal **unowned decisions found**, and the warning **using friendly language to hide missing responsibility**.

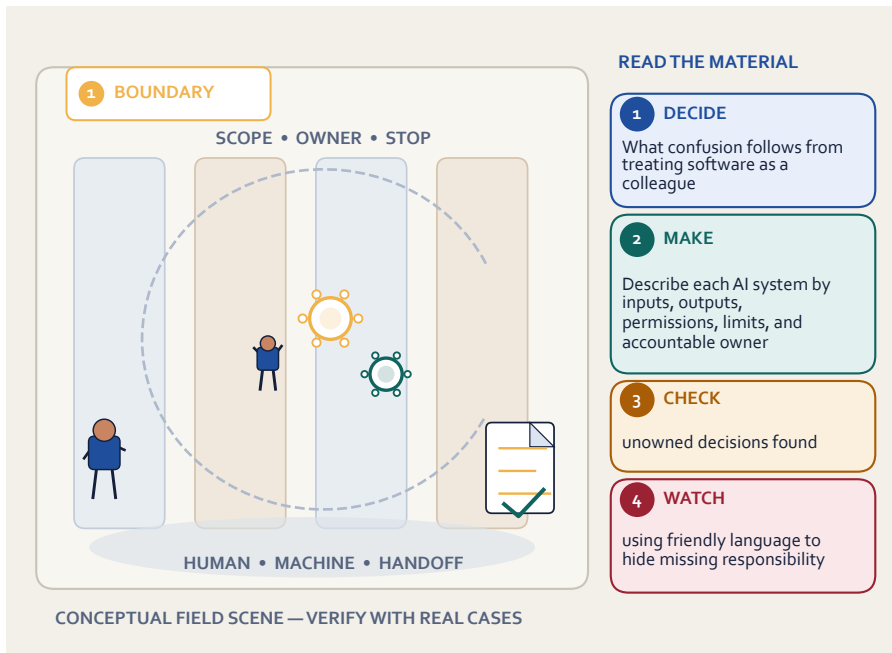


Figure 1.1: Chapter 1 field scene for Stop Calling AI a Teammate Too Soon: a boundary view in the book’s machine team setting.

Read the scene: For Managing the Machine Team, Chapter 1, begin with the central machine team scene and scan DECIDE, MAKE, CHECK, and WATCH. Treat the Stop Calling AI a Teammate Too Soon scene as conceptual, then use its cases and evidence records to test whether unowned decisions found actually improves. For **Stop Calling AI a Teammate Too Soon**, this explanatory plate is a reasoning prompt—not observed evidence or a universal operating prescription.

Visual field note — Stop Calling AI a Teammate Too Soon:

- 1. Locate the tension.** In this composite situation, A marketing team blames a copilot for an unsupported claim that no person reviewed.
- 2. Name the choice.** Answer the chapter question: **What confusion follows from treating software as a colleague?**
- 3. Produce the record.** Use **Describe each AI system by inputs, outputs, permissions, limits, and accountable owner.**, then retain the accountable manager, assumptions, and case evidence needed to review it.
- 4. Test and return.** Track **unowned decisions found**. If **using friendly language to hide missing responsibility**, return to DECIDE and revise the delegation boundary instead of explaining the warning away.

Chapter-specific operating protocol

Phase	Reader action	Evidence retained	Hold condition
1. Define	Write the exact decision behind What confusion follows from treating software as a colleague?	Chapter 1 owner, input, output, exception, and timestamp.	Hold Stop Calling AI a Teammate Too Soon if its record cannot be reconstructed.
2. Contrast	Apply Describe each AI system by inputs, outputs, permissions, limits, and accountable owner to one ordinary case.	Chapter 1 owner, input, output, exception, and timestamp.	Hold Stop Calling AI a Teammate Too Soon if its record cannot be reconstructed.
3. Exclude	Run the same artifact against a counterexample shaped by using friendly language to hide missing responsibility.	Chapter 1 owner, input, output, exception, and timestamp.	Hold Stop Calling AI a Teammate Too Soon if its record cannot be reconstructed.
4. Assign	Name the accountable manager as the person who may continue, narrow, or stop.	Chapter 1 owner, input, output, exception, and timestamp.	Hold Stop Calling AI a Teammate Too Soon if its record cannot be reconstructed.
5. Challenge	Record unowned decisions found and the review effort for the same case.	Chapter 1 owner, input, output, exception, and timestamp.	Hold Stop Calling AI a Teammate Too Soon if its record cannot be reconstructed.

Phase	Reader action	Evidence retained	Hold condition
6. Publish	Release the Stop Calling AI a Teammate Too Soon artifact only when its evidence answers this chapter's question.	Chapter 1 owner, input, output, exception, and timestamp.	Hold Stop Calling AI a Teammate Too Soon if its record cannot be reconstructed.

This sequence is specific to **Stop Calling AI a Teammate Too Soon**. For that Chapter 1 decision, it is not permission to automate; it is a way for managers responsible for teams that use generative AI, copilots, automation, or agents to make the reasoning inspectable at each queue review and team retrospective.

Describe each AI system by inputs, outputs, permissions, limits, and accountable owner. This is the chapter's core production artifact. The **Stop Calling AI a Teammate Too Soon** artifact should fit on one page, use ordinary language, and be reviewed by someone who did not design it. A compact artifact exposes disagreements that a long narrative can conceal.

The chapter's visual tool is the **chapter contract**. For **Stop Calling AI a Teammate Too Soon**, it translates explanation into an inspectable decision aid: the reader can see the decision, artifact, evidence, and exact condition that pauses the shared assignment.

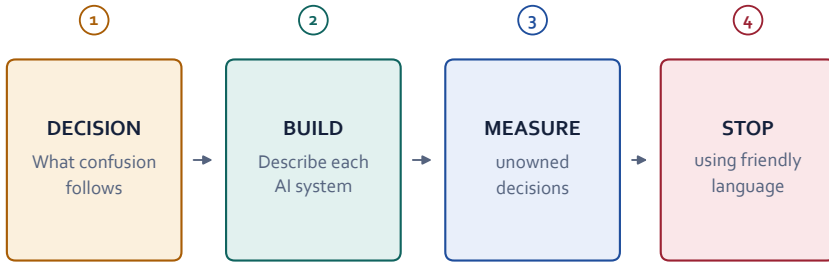


Figure 1.2: Chapter contract connecting the decision, artifact, measure, and stop condition for Stop Calling AI a Teammate Too Soon.

How to use this page: Move across the four Stop Calling AI a Teammate Too Soon cards; compare the decision with unowned decisions found and stop before using friendly language to hide missing responsibility. The chapter-specific measure is **unowned decisions found**; the principal warning is **using friendly language to hide missing responsibility**.

Operating workflow

The **stop calling ai a teammate too soon workflow** makes the chapter executable. For **Stop Calling AI a Teammate Too Soon**, the five stages carry **Describe each AI system by inputs, outputs, permissions, limits, and accountable owner.** from the question **What confusion follows from treating software as a colleague?** to a test against **unowned decisions found**, with **using friendly language to hide missing responsibility** as an explicit return or stop signal.

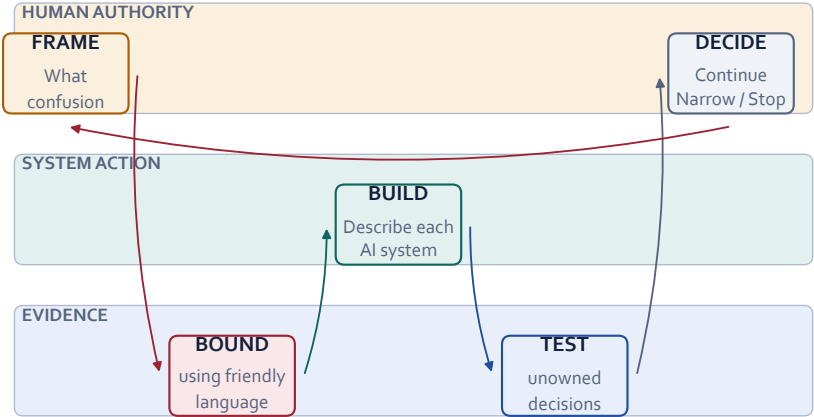


Figure 1.3: Five-stage workflow for chapter 1, Stop Calling AI a Teammate Too Soon: frame the decision, set the delegation boundary, build Describe each AI system by inputs, outputs, permissions, limits, and accountable owner., test against unowned decisions found, then continue, narrow, or stop before using friendly language to hide missing responsibility.

Read the workflow: Move from FRAME to DECIDE for Stop Calling AI a Teammate Too Soon. The lower return path is mandatory when evidence is missing or the stop signal—using friendly language to hide missing responsibility—appears. For **Stop Calling AI a Teammate Too Soon**, record the accountable manager and evidence at every handoff; an arrow is not evidence that the preceding stage passed.

Evidence and control graph

The **stop calling ai a teammate too soon — evidence network** shows the relationships that must remain inspectable when the workflow runs. In *Managing the Machine Team*, this qualitative graph asks the reader to record observed evidence for **unowned decisions found** and route **using friendly language to hide missing responsibility** to recovery instead of treating sample numbers as facts.

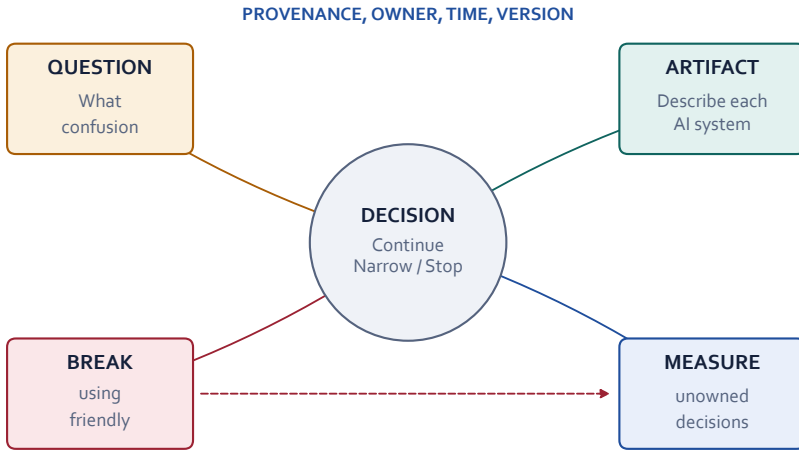


Figure 1.4: Qualitative evidence network for chapter 1, Stop Calling AI a Teammate Too Soon, linking the question to Describe each AI system by inputs, outputs, permissions, limits, and accountable owner., the measure unowned decisions found, and the recovery condition using friendly language to hide missing responsibility without invented empirical values.

Read the graph: Trace the evidence for Stop Calling AI a Teammate Too Soon toward the decision. For Stop Calling AI a Teammate Too Soon, empty or weak evidence does not count as zero risk; it routes the shared assignment to verification, escalation, or stop. Use this **Stop Calling AI a Teammate Too Soon** graph beside **Describe each AI system by inputs, outputs, permissions, limits, and accountable owner**, not as a substitute for recorded cases.

Worked composite case file — illustrative, not empirical evidence

This **Stop Calling AI a Teammate Too Soon** record turns the scenario into an inspectable rehearsal in a human–AI production team. Its numbers are worksheet examples for Chapter 1, not observed results or a prediction of benefit.

Record field	Filled illustrative entry
Situation	A marketing team blames a copilot for an unsupported claim that no person reviewed.
People affected	employees, managers, and customers
Decision	What confusion follows from treating software as a colleague?
Artifact excerpt	Describe each AI system by inputs, outputs, permissions, limits, and accountable owner.
Accountable authority	The accountable manager.
Measure and unit	unowned decisions found — observed events per 100 eligible cases.
Example baseline	17 per 100.
Example target / decision rule	no more than 8 per 100; move only lower, without hiding or excluding hard cases.
Named exception	using friendly language to hide missing responsibility.
Safe default	Return the decision to a named person with sufficient time and context.

Scenario trace

1. **Preserve the starting state.** For **Stop Calling AI a Teammate Too Soon**, record the original request, governing rule, available evidence, and the last safe result before changing anything.
2. **Make the chapter decision.** Answer **What confusion follows from treating software as a colleague?** and record the Chapter 1 assumption that carries the most consequence for employees, managers, and customers.
3. **Use the artifact.** Complete **Describe each AI system by inputs, outputs, permissions, limits, and accountable owner** with a normal case and a countercase; in this chapter, do not infer missing facts merely to finish the form.
4. **Observe the decision rule.** Review **unowned decisions found** at each queue review and team retrospective; the **Stop Calling AI a Teammate Too Soon** record reports exclusions and review effort beside the headline value.

5. **Contain and recover.** If **using friendly language to hide missing responsibility**, return the decision to a named person with sufficient time and context. Preserve the Chapter 1 failed case, assign a deadline, and require the accountable manager to authorize any restart.

Artifact handoff

Start **Stop Calling AI a Teammate Too Soon** from the book-level thesis and boundary. Within *Managing the Machine Team*, attach the current artifact and exception record, then hand both to Chapter 2, **Map Work Into Decisions and Production**, which asks: Which parts of a role are judgment, coordination, or content production?. That recipient may revise the **Stop Calling AI a Teammate Too Soon** assumption, but must preserve the evidence that made this book's earlier decision reasonable.

Evidence application

Reference [1], National Institute of Standards and Technology, *AI Risk Management Framework*, provides a relevant boundary or design input for **Stop Calling AI a Teammate Too Soon**. It does **not** prove that the Chapter 1 worked values, local method, or target will succeed. The **Stop Calling AI a Teammate Too Soon** reviewer must link this book's exact claim to the exact source passage and leave unsupported synthesis marked UNVERIFIED.

Transfer test

- Re-run **Describe each AI system by inputs, outputs, permissions, limits, and accountable owner** with one routine Chapter 1 case that should pass.
- Re-run the **Stop Calling AI a Teammate Too Soon** artifact with missing evidence and keep the safe default active.
- Re-run it where **using friendly language to hide missing responsibility** appears plausible but not certain.
- Ask a reviewer who did not design this chapter's method to decide whether the retained record supports continuing, narrowing, escalating, or stopping.

Measurement and recovery decision

For **Stop Calling AI a Teammate Too Soon**, define **unowned decisions found** as observed events per 100 eligible cases. In this **Stop Calling AI a Teammate Too Soon** worksheet, the Chapter 1 worked baseline is 17 per 100 and the illustrative target is no more than 8 per 100;

both must be replaced with evidence from a human–AI production team. Review the Chapter 1 result at each queue review and team retrospective, preserve excluded and high-consequence cases, and interpret movement as better only when it is lower, without hiding or excluding hard cases.

The named failure is **using friendly language to hide missing responsibility**. Containment means: return the decision to a named person with sufficient time and context. Recovery is complete only after the accountable manager can reconstruct the failed case, explain which assumption changed, and authorize a new test against **unowned decisions found**.

Practice lab

Choose one live but reversible **Stop Calling AI a Teammate Too Soon** example from a human–AI production team. Create the following:

- a one-sentence answer to **What confusion follows from treating software as a colleague?**;
- a completed first draft of **Describe each AI system by inputs, outputs, permissions, limits, and accountable owner**;
- one routine case, one missing-evidence case, and one case shaped by **using friendly language to hide missing responsibility**;
- a locally measured baseline for **unowned decisions found** using the unit observed events per 100 eligible cases;
- a stop threshold, the accountable manager as owner, and the safe default return the decision to a named person with sufficient time and context;
- decision evidence packet that will be handed to the next chapter.

Ask a colleague to challenge **Describe each AI system by inputs, outputs, permissions, limits, and accountable owner**. They should identify one ambiguous field, one person among employees, managers, and customers whose interests are missing, and one reason **unowned decisions found** could reward the wrong behavior. Preserve that challenge in the chapter handoff before revising.

Limits and judgment

Stop Calling AI a Teammate Too Soon supplies a locally testable method, not proof of causal effectiveness or a universal rule. This Chapter 1 method remains inside the book boundary: This book does not determine employment law, promise productivity gains, or treat software as a legal or moral person. Readers applying **Stop Calling AI a Teammate Too Soon** for employees, managers, and customers must verify current law, contract, policy, safeguarding, accessibility, security, and professional duties before consequential use. A negative **unowned decisions found** result may mean

the question was framed badly, the review cost is too high, or return the decision to a named person with sufficient time and context is the responsible decision.

Takeaways

- Metaphors shape expectations, so leaders must separate capability from agency and accountability.
- The chapter artifact is: describe each ai system by inputs, outputs, permissions, limits, and accountable owner.
- The primary measure is **unowned decisions found**, paired with review cost and failure severity.
- Watch especially for **using friendly language to hide missing responsibility**.
- In a human–AI production team, expand scope only after the accountable manager can show the evidence and exercise the recovery path.

Reflection

What evidence would let a skeptical reviewer answer **What confusion follows from treating software as a colleague?** differently from the accountable manager? Which case involving employees, managers, and customers would **unowned decisions found** miss, and would that omission require continuing, narrowing, escalating, or return the decision to a named person with sufficient time and context?