

THE WORKSHOP

Example Mapping

A pattern for breaking a user story into rules and concrete examples in twenty-five minutes, so the team knows whether it's ready to build before the sprint starts. Facilitator's playbook, failure modes, and what to do with the cards afterwards.

2026-04-15

barkingiguana.com/writing/the-workshop-example-mapping/

Contents

Intent	3
Also Known As	3
Motivation	3
Applicability	3
Participants	4
Structure	5
Collaborations	5
Facilitator Playbook	6
Steering When It Goes Sideways	8
Consequences	8
Worked Example	9
Outputs & Follow-up	9
Related Patterns	10

This is the pattern. If you want the story of Example Mapping in action, read [Example Mapping: Making Stories Concrete](#) – the Greenbox team running their first session on the pause-subscription story. This post is the reference you keep open the morning of.

Intent

Break a single user story into rules and concrete examples, in twenty-five minutes, so the team knows whether the story is ready to build and what “done” actually means.

Also Known As

Sometimes called **specification by example** at the sentence level, though specification by example usually implies a longer, heavier process. Sometimes confused with **BDD scenario writing** – Example Mapping produces the material BDD scenarios are written from. Invented and named by Matt Wynne in 2015, originally as a pragmatic answer to the question “how do we make the Three Amigos conversation actually produce something?”

Motivation

A developer picks up a story called “Subscriber can pause their box.” She reads the acceptance criteria, they look fine, she starts building. Three days in she hits a question: what happens to a box that’s already been packed when the pause takes effect? She asks the product owner. The product owner doesn’t know. The product owner asks the warehouse lead. The warehouse lead says “obviously the packed box goes out, we can’t unpack it,” and the developer’s first two days of work are now wrong.

This happens because the story looked simple. It wasn’t. There were three rules hiding inside it and at least one of them depended on operational knowledge nobody had written down. The conversation that would have caught it – a twenty-minute chat between the product owner, a developer, and someone from operations – would have happened before the sprint started if anyone had thought it was worth the time.

Example Mapping exists to make that conversation cheap enough that you always have it. The cards are the forcing function: you can’t hand-wave acceptance criteria when someone asks for a concrete example and you have to write it on a green card.

Applicability

Use when:

- A story is about to enter a sprint and you want to know whether it’s actually ready
- Developers and the product owner suspect they disagree about “done” but haven’t surfaced it
- The story feels simple and you don’t trust the feeling
- You need to decide whether a story should be split, built, or deferred for more discovery

Don't use when:

- The story is genuinely trivial – a copy change, a feature flag flip, a config tweak
- You don't yet know *what* you're building – Example Mapping assumes you have a story, and drills into what it means; use Event Storming or User Story Mapping to get the story in the first place
- The people who know the answers aren't in the room

Inputs. You need one user story written on a yellow card and placed at the top of the table. No other preparation is required – participants who have read the story in advance are a small bonus, not a prerequisite.

Levels. Example Mapping doesn't have formal levels the way Event Storming does, but in practice there are two modes, and the mode changes the shape of the session. Start at Story Level unless you have a specific reason not to.

Level	Scope	Duration	Output
Story Level (<i>default</i>)	One user story about to enter a sprint	25 minutes	Rules, examples, questions, build/split/defer verdict
Epic Level (<i>zoom out</i>)	A cluster of related stories in an epic	90 minutes, multiple passes	Rough split of the epic into buildable stories

Story Level is what most teams actually need. One story, one table, twenty-five minutes. Zoom out to Epic Level only when you're trying to decide how to break up a large feature area, and even then, Epic Level is really three or four Story Level sessions stacked together with a break in between.

Participants

Facilitator. Runs the clock, asks for the next example, keeps people off implementation detours. Does not put cards on the table themselves except to demonstrate the colour convention at the start.

Product owner. Mandatory. They own the story and they're the person who decides which rule applies when two participants disagree. If the product owner can't attend, reschedule. A session without them is theatre – the rules will get written, but nobody in the room will have the authority to say they're right.

Developers. At least one, ideally two. They're the ones who will catch the rules that are impossible or expensive to implement as written, and they're the ones who benefit most from leaving the session with concrete tests in hand.

Tester or QA. Highly valuable if you have one. They will think of edge cases faster than anyone else in the room and they are the natural customer for the green cards.

Operations, support, or domain expert. When the story touches a part of the system only one person really understands – the warehouse lead, the SRE who owns the cron that matters, the support agent who talks to the subscribers who hit this particular path – pull them in for this one session. They'll save you a week.

Group size: 3–5 including the facilitator. This is not a whole-team workshop. Fewer than three and you don't get enough friction between perspectives; more than five and the table becomes a meeting.

Who to leave out:

- **The rest of the team.** They don't need to be there. They'll get the output through the acceptance criteria.
- **Stakeholders who want to observe.** Observers warp the conversation. If they care about the story, they should come as participants or read the output afterwards.
- **Senior leaders who won't stop rewriting the rule cards.** Their job happens before or after Example Mapping, not during.

Structure

Phase	Duration	Cards	Key question
Explain the colours, read the story	2 min	Yellow	"What are we mapping?"
Seed the first rule	3 min	Blue + green	"What's the most obvious rule? Give me an example."
Explore the rules	15 min	Blue + green + red	"What happens if...?"
Assess readiness	5 min	Review all	"Is this ready to build?"
Total	25 minutes		

Twenty-five minutes is the whole pattern. The time constraint is not cosmetic. If you can't map the story in twenty-five minutes, the story is telling you something: it's too big, too vague, or standing on unresolved questions. That signal is worth the entire session by itself.

Collaborations

Four card colours, and each colour belongs to a role in the conversation:

- **Yellow** – the story. One card, at the top of the table, present for the whole session.
- **Blue** – rules. Acceptance criteria phrased as business rules. "A subscriber can pause for up to eight weeks."
- **Green** – examples. Concrete scenarios that illustrate a rule. "A subscriber pauses on a Monday for two weeks. Her Wednesday box skips. Her next box is the following Wednesday."
- **Red** – questions. Things nobody in the room can answer. "What happens to a box that's already been packed when the pause takes effect?"

The rhythm is: someone proposes a rule, someone else asks for an example, the example either confirms the rule or reveals a case the rule didn't cover, which produces another rule or a red card. Blue cards stretch left-to-right across the table in a row under the story; green cards drop downward in columns under the rule they illustrate; red cards go off to the side where they can be counted at the end.

The whole thing runs as an open conversation. No silent writing phase. The facilitator's role is to keep asking the next question – "can someone give me a concrete example of that?" – until the rule is understood or a red card goes down.

Facilitator Playbook

Phase 1 – Seed the rules (3 minutes)

Read the yellow card aloud. All of it. Don't paraphrase. If the story is two sentences long, read both sentences.

Then set the colour convention:

"Yellow is the story – it's already on the table. Blue is for rules. A rule is an acceptance criterion phrased as a business rule: 'a subscriber can pause for up to eight weeks.' Green is for examples. An example is a concrete scenario: 'A subscriber pauses on a Monday for two weeks, her Wednesday box skips.' Red is for questions nobody in this room can answer right now. We'll come back to those at the end."

Now draw out the first rule. The product owner almost always has the obvious one:

"What's the most basic rule – the thing that has to be true for this story to exist at all?"

Write it on a blue card. Place it below the yellow story. Then immediately:

"Can someone give me a concrete example of that rule? One specific scenario. Real names are fine."

Write the example on a green card. Place it under the blue rule. You now have the shape of the session: yellow on top, blue in a row underneath, green in a column under each blue. Once the shape is visible, the room knows what to do.

What to watch for:

- **Abstract examples.** *"The subscriber pauses and it works"* is not an example. Push gently: *"Give me names and dates. What day do they pause? What happens to their next box?"*
- **Rules that are actually two rules.** *"A subscriber can pause and unpauses their box"* is two rules. Split them: *"Let's put pausing on this card and unpausing on a new one."*
- **Implementation leaking in.** *"We'd need a cron job to check..."* – redirect: *"Park the cron. Let's nail down what should happen first, then worry about how we build it."*

Phase 2 – Explore the rules (15 minutes)

This is the core of the session. You now cycle: next rule, example, next rule, example, red card, next rule, example. The facilitator's job is to keep the cycle moving by asking one of four questions over and over:

"Can someone give me an example of that?"

"What happens if...?"

"Is that always true, or only sometimes?"

"Is that the same rule or a different rule?"

The conversation will naturally surface new rules as people think of edge cases. A green card someone writes to illustrate one rule will turn out to break a different rule, which produces a new rule, which needs its own examples. That's the session working. Let it flow.

Place blue cards in a row. Place green cards in columns under their rule. Place red cards off to the right – a visible pile the team can count.

What to watch for:

- **Too many rules.** If you're past six or seven blue cards and still producing more, the story is too big. Note it; you'll address it in Phase 3.
- **Rules with no examples.** A rule nobody can illustrate with a concrete scenario is either trivially obvious (fine, move on) or poorly understood (not fine, red card it). Don't let a rule sit without an example.
- **Productive disagreement.** Two participants disagree about what should happen in a specific example. Capture both versions as green cards. The product owner decides which one the business actually wants, or red-cards it if they don't know.
- **Red cards piling up.** More than three or four red cards means the story isn't ready. That's valuable information, not a failure. Keep going to the end of the session – the fuller picture is worth the remaining minutes.

Example Mapping works the same for infrastructure stories, with the SRE or pipeline owner taking the product owner's seat. A rule might be *"the pipeline rolls back on failed health checks"* and an example *"health check fails at 10:02, rollback begins at 10:03, service back on previous version by 10:05."* Same shape, different domain.

Phase 3 – Assess readiness (5 minutes)

Step back from the table. Look at the shape of the cards. The shape tells you the verdict.

- **Few blue cards, examples for each, no red cards** – the story is well understood. Build it.
- **Many blue cards, examples for each, no red cards** – the story is well understood but too big. Split it, probably by rule.
- **Any red cards** – the story has open questions. Assign each red card to a person with a date. Don't commit to building until they're answered.
- **Very few cards, session finished in twelve minutes** – either the story really is trivial, or the team is being superficial. Probe once: *"Is there any scenario we haven't considered where this would behave differently?"* If the answer is a confident no, you're done.

State the verdict out loud. Someone – ideally the product owner – says the phrase: "This is ready to build" or "This needs to be split" or "This isn't ready until these red cards are answered." Saying it out loud matters. It's the commitment, and it's what everyone remembers when someone later asks "wait, did we decide about this?"

What to watch for:

- **Declaring "it's fine" too quickly.** If there are red cards, it's not fine. If there are eight rules, it's probably not one story.

- **Trying to split the story on the spot.** Splitting is a ten-minute conversation at minimum and it belongs in a separate session. Note the suggestion, flag the story for splitting, and move on.
- **The product owner deferring to the developers.** The verdict is theirs to state. If they look to the developers for the “ready or not” call, ask them directly: *“Product owner, what do you want to do with this story?”*

Steering When It Goes Sideways

The monologue. The product owner explains the story for ten minutes while everyone listens politely.

Recovery: Interrupt and cash the monologue in for cards: *“Stop there. Can someone capture what they just said as a rule on a blue card?”* Forcing people to write cards forces them to be precise. *Stop if:* The monologue restarts after a second redirect. The story isn’t ready for Example Mapping – it needs a longer conversation first, and you should schedule one.

The rabbit hole. The team is fifteen minutes into debating one edge case. *Recovery:* Cash it in as a red card: *“This is a great question. Let’s put it on red and keep moving. We’ll come back to it or take it offline.”*

Stop if: The same edge case keeps surfacing after two red cards. There’s a deeper unknown and you need a time-boxed investigation outside the room before another Example Mapping session will land.

The empty table. Nobody is writing cards. The conversation is circular. *Recovery:* The story is too vague. Ask a sharply concrete question: *“What’s the simplest version of this story? One subscriber, one action, one outcome. What happens?”* If that produces a green card, you have a seed. *Stop if:* The room genuinely can’t answer the simplest version. The story needs discovery, not mapping.

The feature creep. Someone keeps adding rules that belong to a different story. *Recovery:* *“That sounds like a different story. Let me grab a yellow card and we’ll park it next to this one.”* A small pile of parked yellows is a signal to the product owner. *Stop if:* Half the rules being proposed belong to other stories. Something is structurally wrong with how the story was written. Stop and reshape it.

The premature solution. Developers start debating database schemas and API signatures. *Recovery:* *“Great implementation thinking – hold it for when you start the story. Right now we’re still mapping what should happen.”* *Stop if:* They can’t hold the distinction after three prompts. They’ll get more value from a design session.

The silent developer. A developer is in the room but not speaking, not writing, not asking for examples.

Recovery: Name them and ask directly: *“What’s the part of this story you’re most worried about implementing?”* Worry is the fastest route to a red card. *Stop if:* They disengage completely. Something else is going on. Don’t try to fix it in-session.

Consequences

Benefits

- Acceptance criteria concrete enough to write tests from directly – each green card is almost a BDD scenario
- A clear, stated verdict on whether the story is ready, needs splitting, or needs more discovery
- Red cards captured with owners, not left as vague worries

- Misunderstandings about “done” caught on cards before a developer picks the story up

Costs

- 75–150 person-minutes per story, for a 25-minute session with 3–5 people
- The product owner has to be available for every session
- Run consistently, it’s one more meeting on the calendar – though usually a shorter one than the conversations it prevents

Failure modes

- The session runs long because the story was too big and the team tried to force-finish instead of stopping
- Red cards get ignored afterwards and the story goes into the sprint anyway
- The product owner accepts the developers’ verdict rather than making their own
- Rules get written without examples and nobody notices

Stop signals

- Twenty minutes in and the table is still mostly empty
- The product owner is absent or checking their phone
- Every rule is producing a red card – you’re not mapping, you’re discovering, and that’s a different session

Stopping early when the signal is clear is not failure. Running a doomed session to the 25-minute bell is.

Worked Example

See [Example Mapping: Making Stories Concrete](#) for the Greenbox team running their first session on the pause-subscription story – including the moment a green card about an already-packed box turns into the red card that reshapes a week of work.

Outputs & Follow-up

Facilitator’s close-out (same day)

- Photograph the card layout from directly above. One clear shot is enough – the grid is small.
- Transcribe the rules into the story’s acceptance criteria in the tracker.
- Transcribe the green cards as test scenarios, in BDD format if the team uses it.
- Write the red cards into the tracker as open questions, each with an assigned owner and a date.

The product owner’s week

This is where the pattern earns its cost, and the work is mostly the product owner’s.

- **Chase every red card to resolution.** Each red card is an open question with a name attached. The product owner’s job is to make sure those conversations happen before the story enters a sprint. A red card ignored is a production bug rehearsed.

-
- **State the verdict in writing.** Update the story with “Ready to build,” “Needs splitting,” or “Blocked on [red cards].” The verdict is the single most valuable artefact from the session.
 - **If the story needs splitting, split it.** Each rule with its examples can often become its own story. Don’t defer the split until sprint planning – the shape is freshest now.
 - **Walk the acceptance criteria back to the developers.** They were in the room, but the transcribed criteria may look different from what they remember. Five minutes of “does this match what we decided?” prevents the slow drift between session and sprint.

Ongoing

- Run Example Mapping for every story before it enters a sprint. It takes twenty-five minutes and consistently prevents the mid-sprint “but I thought it meant...” conversations that cost days.
- Track the red card rate. If it’s trending upward, your stories are arriving at Example Mapping too raw – push for better discovery upstream.
- Keep the green cards visible during the sprint. They’re the tests, and having them on the team board keeps “done” honest.

Related Patterns

- **Event Storming** – hotspots from an Event Storming session often become the input to Example Mapping. Event Storming finds the questions; Example Mapping turns them into rules and examples.
- **User Story Mapping** – Story Mapping produces the backlog of stories; Example Mapping decides whether each of those stories is ready to build.
- **Sprint Planning** – Example Mapping is the gate before sprint planning. A story that hasn’t been through Example Mapping shouldn’t be in the sprint conversation.
- **Decision Tables** – when a rule has many conditions and combinations, the green cards become unmanageable. That’s a signal to promote the rule to a decision table.
- **Assumption Mapping** – red cards that can’t be answered by anyone in the room are often assumptions in disguise. They belong on the grid.

About this playbook

This playbook is part of *The Workshop*, a reference series of facilitator playbooks published at barkingiguana.com. The canonical, up-to-date version lives at barkingiguana.com/writing/the-workshop-example-mapping/.

These posts are LLM-aided. Backbone, original writing, and structure by Craig. Research and editing by Craig + LLM. Proof-reading by Craig.