

RDD Workbench: Workshop Guide

© 2026 Codemanship Ltd.

Table of Contents

RDD Workbench: Workshop Guide 1

The process at a glance 4

1. Load your feature files 5

2. Assign responsibilities 6

3. Name the roles 8

4. Refine the CRC cards 9

5. Walk through scenarios to find the collaborators 10

6. Review the class diagram 12

7. Review the design 13

Saving and sharing your model 15

Quick reference 16

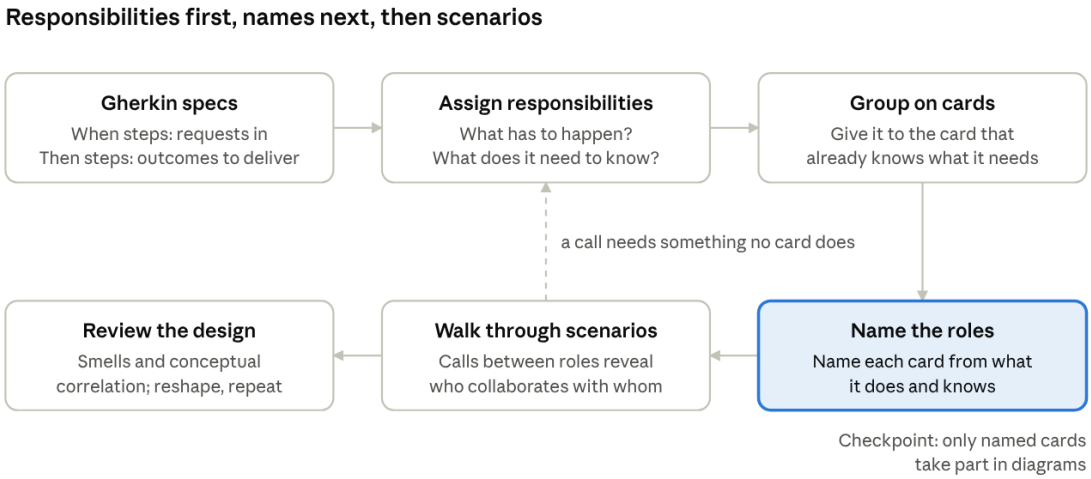
RDD Workbench Guide: Responsibilities First

Oct 20, 2026 · @Jason

In Responsibility-Driven Design you start with what has to happen, not with who does it. This guide covers the responsibilities-first version of the RDD Workbench: you put each outcome in your Gherkin specs on a card as a responsibility, name the cards once you can see what they do, then walk through scenarios to find out who works with whom.

The process at a glance

You find the design by asking what has to happen before asking who does it. Names come once each card shows what it is for, and collaborations come from acting out scenarios, not from guessing.



The RDD process in the workbench · 6 stages

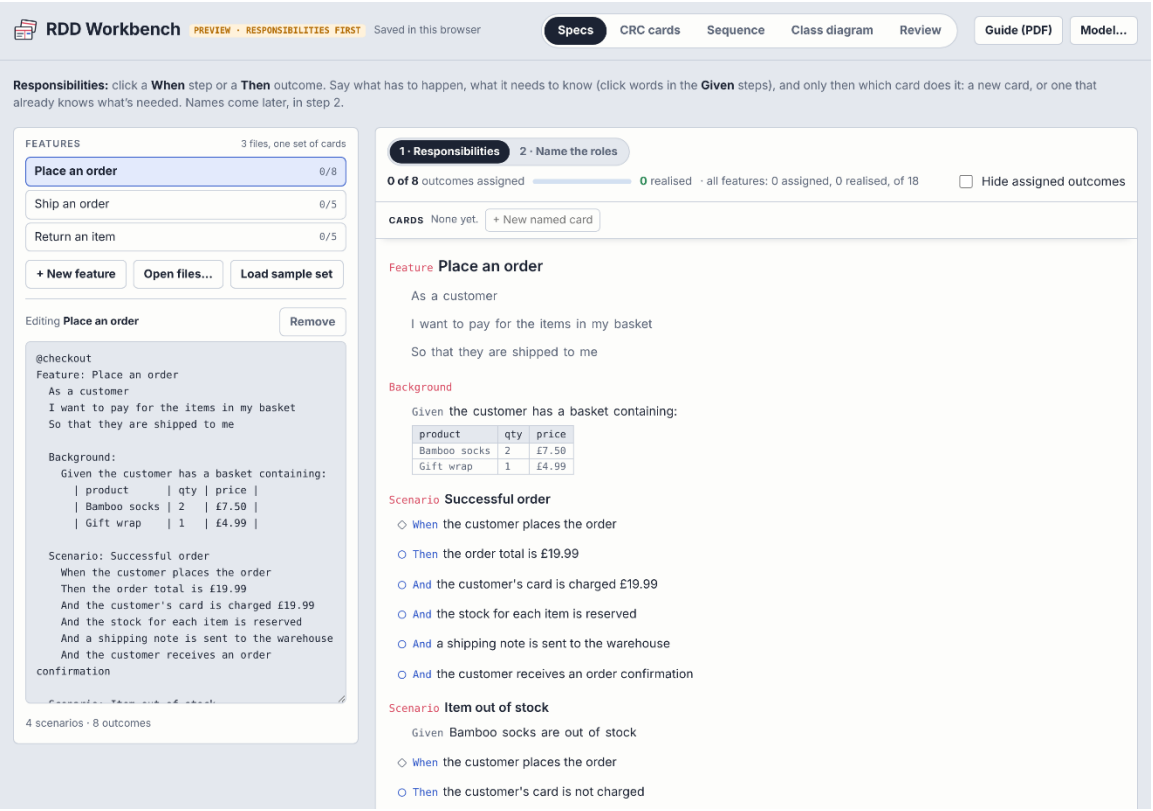
1. **Start from the specs.** Each **When** step is a request into the system. Each **Then** step is an outcome the system must deliver. Both become responsibilities: something that will be done.
2. **Assign responsibilities.** For each step, say what has to happen ("Calculate order total") and what it needs to know to do it (products, prices, discounts). At this point you don't know who does it, and you don't need to. *Something* will.
3. **Group them on cards.** Put the responsibility on a new, unnamed card, or on a card that already knows what it needs. This is Information Expert: the job goes to whoever has the information. Each card gathers a few things it does and the things it knows to do them, as on a blank index card in a face-to-face workshop.
4. **Name the roles.** Once a card is full enough to show its purpose, find the best name for it, preferably in the language of the spec. Naming is the checkpoint: you can't act out a scenario with a card nobody has named, so only named cards go into sequence and class diagrams.
5. **Walk through scenarios.** Act out each scenario as calls between roles. Each call asks another role for one of its responsibilities, and that makes the caller and receiver collaborators. When a call needs something no card does yet, go back and add it.

6. **Review the design.** Look for smells (too many responsibilities or dependencies, cycles, envy) and check how closely the design's words match the spec's. Reshape, rename, and walk the scenarios again.

The workbench's tabs follow the same order: **Specs** (stages 1 to 4), **CRC cards**, **Sequence**, **Class diagram** and **Review**.

1. Load your feature files

Start on the **Specs** tab. The left-hand panel lists your feature files; the right-hand panel shows the one you're working on. The workbench opens empty, ready for your own features; the sample set is there if you want to practise first.



- **Load sample set** loads three related features to practise on: Place an order, Ship an order and Return an item.
- **Open files…** loads `.feature` files from your computer. You can pick several at once, or drag them onto the panel. A file with the same name as one already loaded replaces it.
- **+ New feature** gives you an empty feature to type or paste Gherkin into.
- Click a feature in the list to work on it. The numbers beside it (for example 5/8) show how many of its outcomes you've assigned.
- **Remove** takes the feature you're editing out of the model. Anything already on the cards stays.

Each feature needs its own `Feature:` name. If two files share one, you'll see a warning under the text box. Related features share one set of cards, so a card built for one feature is there to reuse in the next.

2. Assign responsibilities

The Specs tab works in two steps, chosen with the switch at the top of the spec: **1 · Responsibilities** and **2 · Name the roles**. It opens in step 1. Here you go through the spec line by line, saying what has to happen and putting it on a card. Nobody has a name yet.

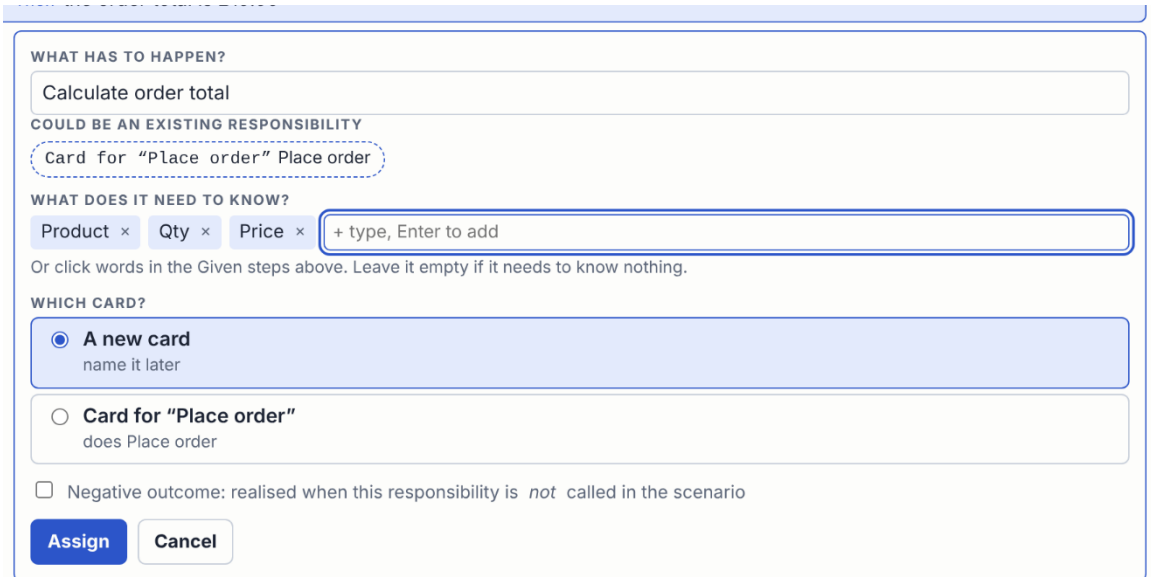
What you click decides what you add:

You click	It becomes	Counts towards progress
A **When** step	Something a card does, asked for by the Client	No
A **Then** step, or an **And** or **But** after it (an outcome)	Something a card does	Yes
Words in a **Given** step, or a table heading, while no form is open	Something a card knows	No

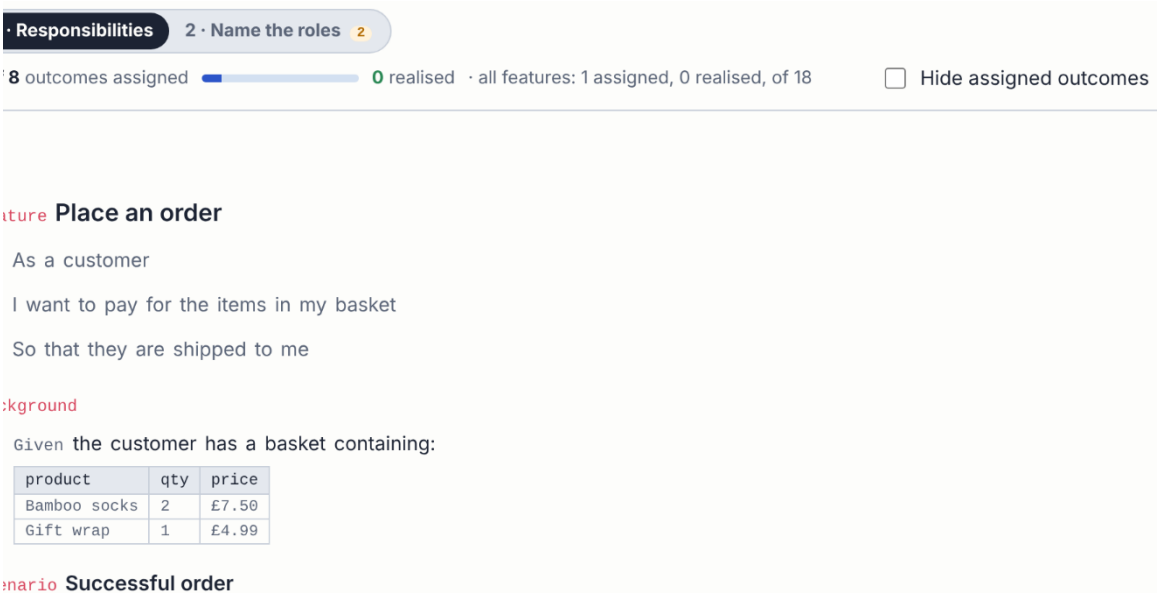
Assigning a When step or an outcome. Click it, and a form opens beneath it with three questions, in order:

1. **What has to happen?** Name the responsibility as something done, such as "Calculate order total". If it could be one you already have, it's suggested underneath; click a suggestion to use it.
2. **What does it need to know?** Add the information this responsibility uses. Click a suggested chip (taken from table headings in the Given steps, including the Background, and from things cards already know that the scenario mentions), click or drag across words in a Given step, or type and press Enter. Leave it empty if it needs to know nothing.
3. **Which card? A new card** is chosen unless you pick another. Below it are the cards you already have, each described by what it does and knows. Cards that already know what you said it needs are listed first and marked, for example **knows Price**. Giving the job to the card that has the information is Information Expert.

Then click **Assign**. After an outcome, the workbench moves on to the next unassigned one, ready for the next responsibility.



What it needs is added to the card as things it Knows, and remembered against that responsibility, so the card shows *needs Product, Qty, Price* under it.



When steps are entry points. A When step is a request into the system, so the card you give it is the one the Client calls first in that scenario's sequence diagram. The same When step often opens several scenarios; the form offers to assign all the identically worded ones in the feature at once, ticked by default. When steps show a diamond marker and an **entry** tag, and don't count towards progress.

Reuse a responsibility whenever it fits. One responsibility often delivers several outcomes, across scenarios and features. Pick it from the suggestions, or choose the card that already does it (it's marked **already does this**).

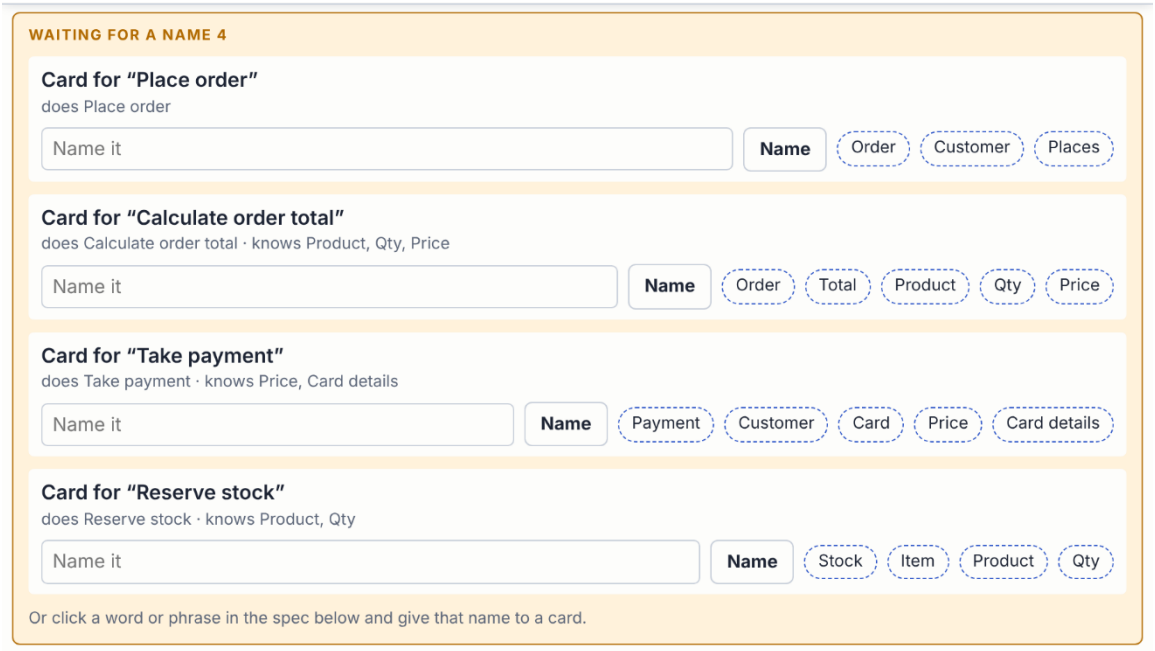
Negative outcome is optional. Tick it when a Then says something *doesn't* happen, such as "the card is not charged". The chip then shows **NOT** before the responsibility, and the outcome counts as realised when the scenario's sequence diagram does *not* call it. Click ¬ on a chip to switch it on or off later.

Assigned lines show a chip with the card and responsibility. A card with no name is shown by its first responsibility, such as *Card for "Calculate order total"*. A **+2** on the chip means the same responsibility is linked to two other lines; hover to see which. Click **×** on a chip to unassign it, or click an assigned line again to give it to a second card as well. Tick **Hide assigned outcomes** to see only what's left to do.

To add something a card knows without an outcome, close any open form and click words in a Given step or a table heading, then choose the card.

3. Name the roles

Once a card holds enough to show what it's for, give it a name. Switch to **2 · Name the roles**; the number on the switch is how many cards are still waiting for one. You can name cards at any time, and go back to step 1 whenever you like.



Waiting for a name lists every card without one, described by what it does and knows. For each card:

- Type a name and press Enter, or click **Name**.
- Or click one of the suggestions after the button. They're the words that come up most in the card's outcomes, its responsibilities and what it knows, so they come from the spec.

Names stay as you'd say them: "shipping note" becomes "Shipping note". Code-style names such as `ShippingNote` appear only in the class diagram. Two cards can't have the same name; if they turn out to be the same role, merge their items onto one card instead.

Naming from the spec. Click any word in the spec below the panel: the feature name, its description ("As a customer…"), a scenario name, any step, or a table heading. For a phrase, drag across the words, or click the first word and then the last. A form opens with the name. Under **Give this name to**, click the

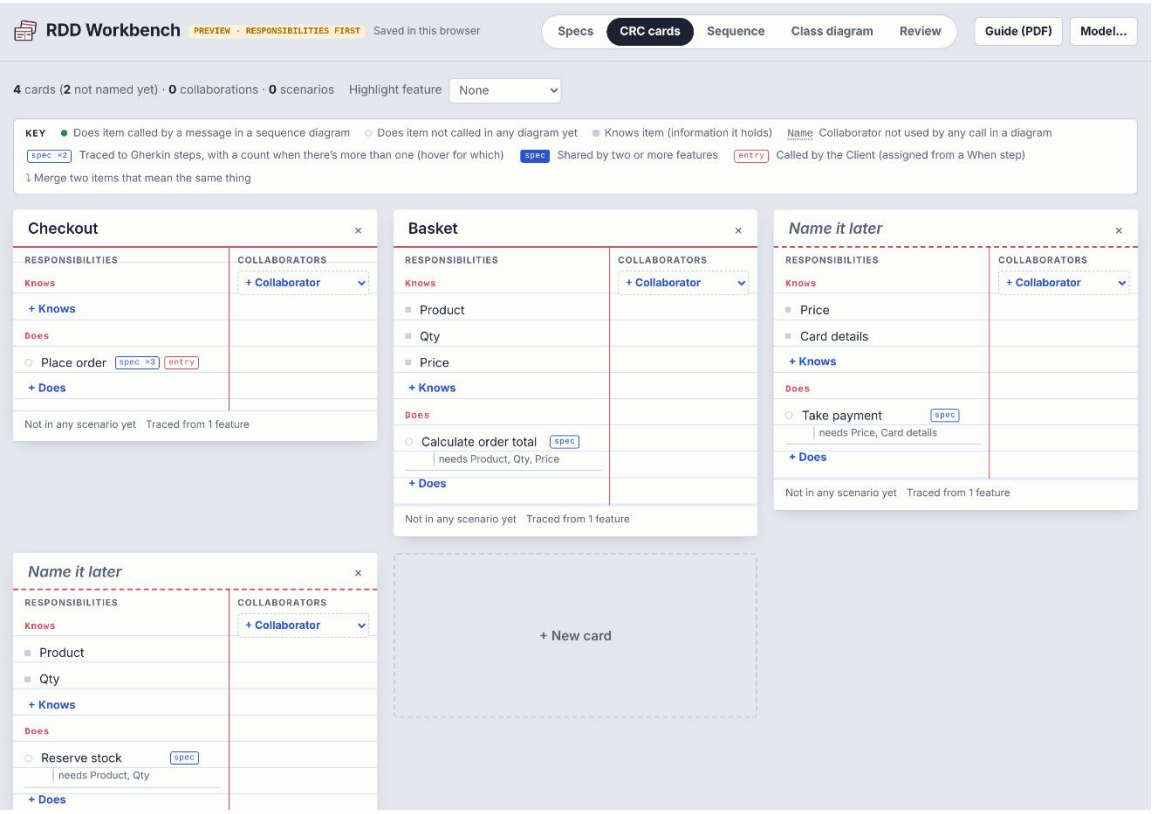
card it names. Or click **Make it a new card** for a role that has no card yet, or **Link to …** when a card already has that name. Clicking a card in the **Cards** tray does the same.

Phrases you've made into names are underlined in red, and every other mention of a card's name gets a dotted red underline, so you can see who the spec talks about. When nothing is selected, click a card in the tray to highlight where it appears in the feature: every mention, and its assignments, get a blue background. **Next** steps through them; click the card again, press Escape or click **Clear** to stop.

A good name is one the people who wrote the spec would recognise. The Review tab (section 7) shows how many of your words come from the spec.

4. Refine the CRC cards

The **CRC cards** tab shows every card: what it **Knows**, what it **Does**, and its **Collaborators**, the roles it asks for help. The line at the top counts the cards still to be named.



- **Cards without a name** have a dashed line under the title and say *Name it later*. Click the title and type to name one here instead of in the Specs tab.
- **Needs:** under a responsibility, *needs Product, Qty, Price* lists what you said it needs to know. Renaming one of those Knows items, or the responsibility, keeps the list up to date.
- **Edit anything in place.** Click a name or item and type. Press Enter when you've finished typing; use + Knows or + Does to add another item. Renaming an item keeps its links to the specs, and renaming a Does item also renames every call to it in the sequence diagrams.
- **+ Knows**, **+ Does** and **+ Collaborator** add to a card. Hover over an item and click **×** to remove it. **+ New card** adds a card.

- **Hover over a card** to highlight the cards it works with. Click a collaborator's name to jump to its card.
- **Highlight feature** (top of the tab) shows which cards and items one feature needs, and dims the rest.

Collaborators mostly come from the sequence diagrams (section 5), but you can add them by hand. A dotted name isn't used by any call yet. Removing one that calls still use asks you to confirm first.

Multiplicity: the small field after each collaborator says how many of them the role works with. A grey **1** is the default. Type another value, such as **0..1**, ***** or **1..***, to change it, or clear it to go back to 1. It's the same value as the multiplicity at that end of the class diagram, so changing either changes both.

Types: when several roles are played by the same kind of object, such as a customer and a manager who are both people, hover a card and click **+ Type**. Enter the type and press Enter. The card then shows "played by **Person**". Roles of the same type share one class in the class diagram (section 6). Everything else, including collaborators and messages, stays with the roles.

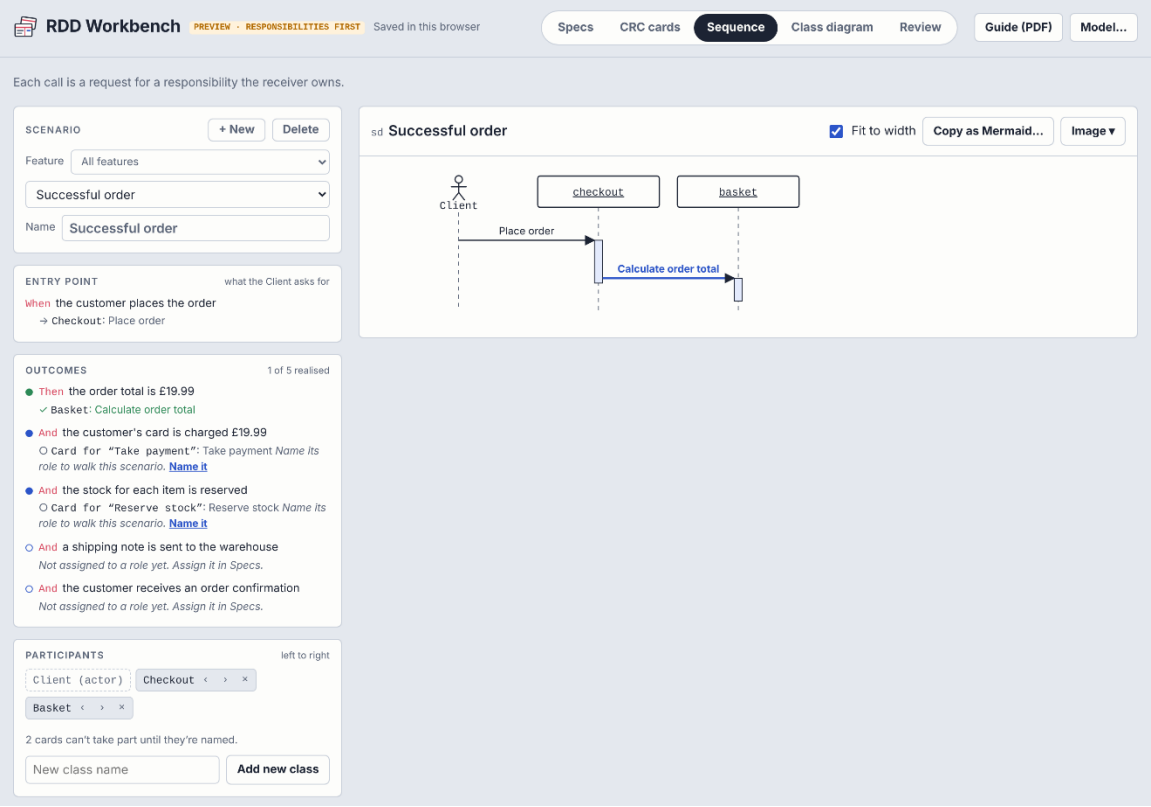
Merging: when two items turn out to mean the same thing, hover over one and click ⤳, then choose the item to merge it into. Its spec links and sequence messages move with it.

Deleting a whole card also deletes its messages in every sequence diagram, plus any message that depended on it, so no diagram is left with stranded calls. The delete button shows how many messages will go before you confirm. If you delete an item that was linked to an outcome, the Specs tab marks that link and offers **Re-add to card**.

5. Walk through scenarios to find the collaborators

The **Sequence** tab tests the cards by acting out a scenario as messages between roles. Each call asks the receiving role to do one of its responsibilities, and that is how collaborations are found: once a call asks for a responsibility, the receiver becomes a collaborator of the sender. The diagram on the right redraws as you edit.

Only named cards take part. You can't act out a scenario with a card nobody has named, so cards without a name aren't offered as participants, and the panel says how many are waiting. In the **Outcomes** panel, an outcome given to a card without a name says *Name its role to walk this scenario*; click **Name it** to open the card.



1. **Scenario:** choose a feature, then one of its scenarios. Scenarios marked "not drawn yet" come from that feature's Gherkin; choosing one starts its diagram, already holding the Client's calls for any When steps you've assigned to named cards. Or click **+ New** for a scenario of your own.
2. **Participants:** every diagram starts with **Client**, a stick figure standing for whoever asks the first role to act. Add the roles involved, left to right, with **+ Add a card…**. Type a name and click **Add new class** for a role that has no card yet.
3. **Messages:** click **+ Message** to add one at the end, or ┼ on any message to insert one straight after it. Pick the sender and receiver, and choose **→ call** or **⇠ return**. A call asks the receiver for one of its responsibilities: choose it from the receiver's Does items, or pick **┼ New responsibility for …** to add one to its card on the spot. Name a return after what comes back. Use the arrows to reorder.

Each role's lifeline is labelled the UML way, as an underlined instance name starting with a lower-case letter: **checkout**, or **customer : Person** when the role has a type.

Who sends it. A new message is sent by whoever holds the baton at that point. A call passes the baton to its receiver, and a return hands it back. **┼**, **+ Message** and **+ Add call** all use this rule, so change the sender only when someone else should ask.

The Outcomes panel lists the scenario's Then steps and ticks each one off as the diagram realises it. Click **+ Add call** beside one that isn't realised yet to add the call it needs. If the diagram already ends with returns to the Client, the call goes in before them. **The Entry point panel** shows the scenario's When steps and their cards; click **+ Add Client call** if the Client doesn't make one of those calls yet.

Returns. Click ⇠ on a call to add its return: **Next message** puts it straight after the call, **Last message** at the end of the diagram. If the receiver makes calls of its own first, a third option puts the return after those. The button only shows on calls that still need a return, so it also shows which are missing.

Repeating calls. When a role works with many of a collaborator (a multiplicity such as ***** or **1..***), its calls to that collaborator show ↻. Click it to repeat the call for each one. The diagram draws a UML loop frame around the call, its return and anything the receiver does in between; calls straight after it to the same receiver join the loop too. Above the first message in the loop, edit the condition ("for each line item"), click **Include next** or **Leave out last** to resize it, or **×** to remove it.

Selecting. Click a message in the list, or its arrow in the diagram, to select it. Both stay highlighted in blue, and each side scrolls to show the other.

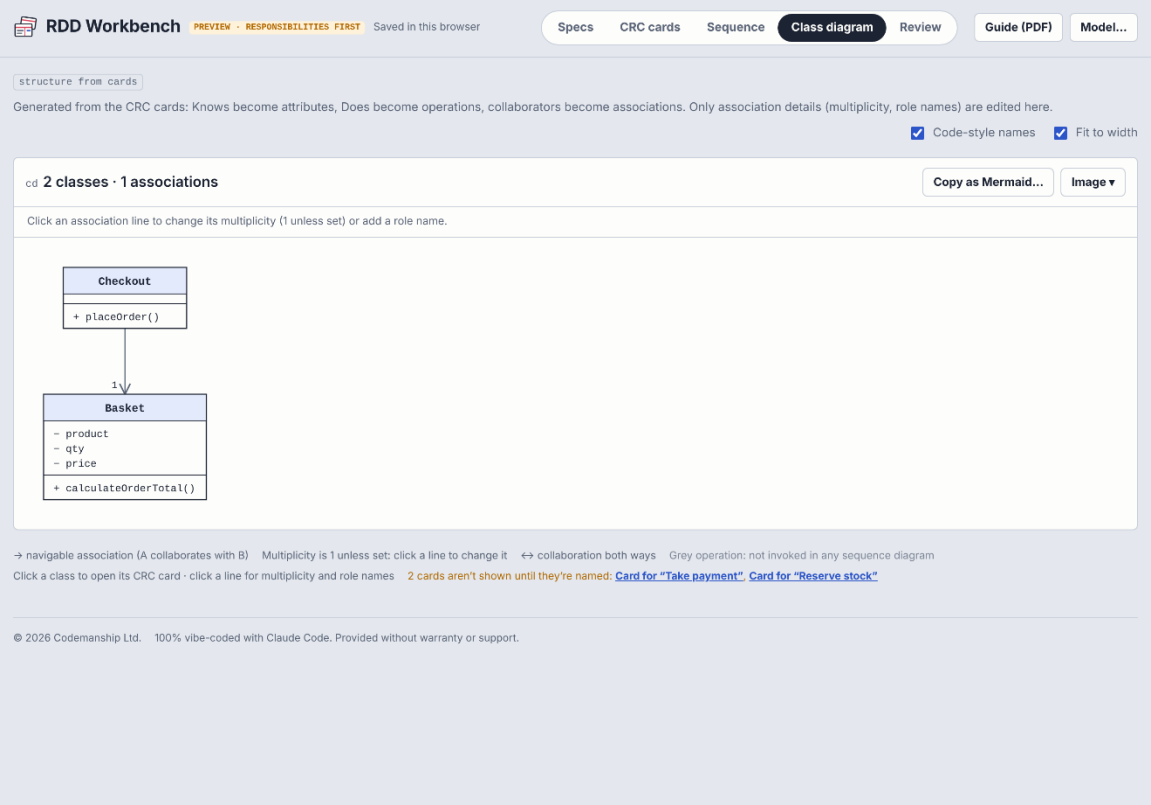
Unreachable messages. A message drawn in dotted orange has a sender that hasn't been called earlier in the scenario, so nothing asks it to act. Move a call to that role above it, or delete the message. Unreachable messages don't count towards realising outcomes.

Collaborations. Returns, and calls with no responsibility chosen yet, never add collaborators. A message can't ask for something on a Knows list: roles do things, they aren't queried for data. If a change leaves a collaboration unused in every scenario, such as correcting a wrong sender or deleting a message, a prompt asks whether to delete it: **Delete it** or **Keep** (**Delete all** or **Keep all** when there are several).

Fit to width shrinks the diagram to the panel. **Copy as Mermaid…** gives you the diagram as text for GitHub, Notion or slides. **Image** ▾ copies it as a picture or saves it as an SVG or PNG file.

6. Review the class diagram

The **Class diagram** tab draws the structure your named cards add up to. Knows items become attributes, Does items become operations, and each collaborator becomes an arrow towards the role it calls. You change the design on the cards; this tab only shows it.

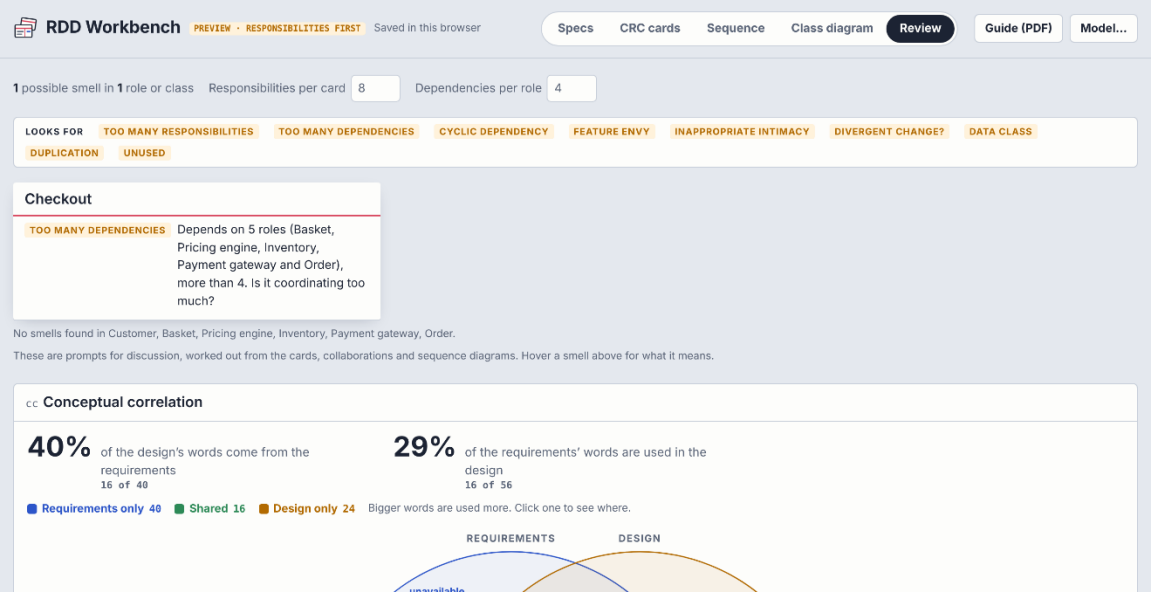


- **Cards without a name aren't drawn.** A class needs a name, so they're listed under the diagram instead ("2 cards aren't shown until they're named"), each linked to its card. The diagram fills in as you name them.
- **Grey operations** aren't called in any sequence diagram yet. That's worth a question: is the responsibility needed, or is a scenario missing?
- **Click an association line** to change the multiplicity at each end (for example 1, 0..1 or *) and add a role name at the end the arrow points to. It's the same value as on the CRC card's collaborator. Click **Done** to close it.
- **Click a class** to open its CRC card.
- **Code-style names** switches between names as written ("Take payment") and as code (`takePayment()`). **Fit to width** shrinks a large diagram to the screen.
- **Copy as Mermaid…** gives you the diagram as text, including multiplicities and role names. **Image** ▾ copies it as a picture or saves it as an SVG or PNG file.

Types. Roles of the same type (section 4) share one class, with their attributes and operations combined. Each pair of collaborating roles still gets its own line, and a role's name appears at its end of a line when it differs from the type there. Clicking a shared class lists the roles that play it.

7. Review the design

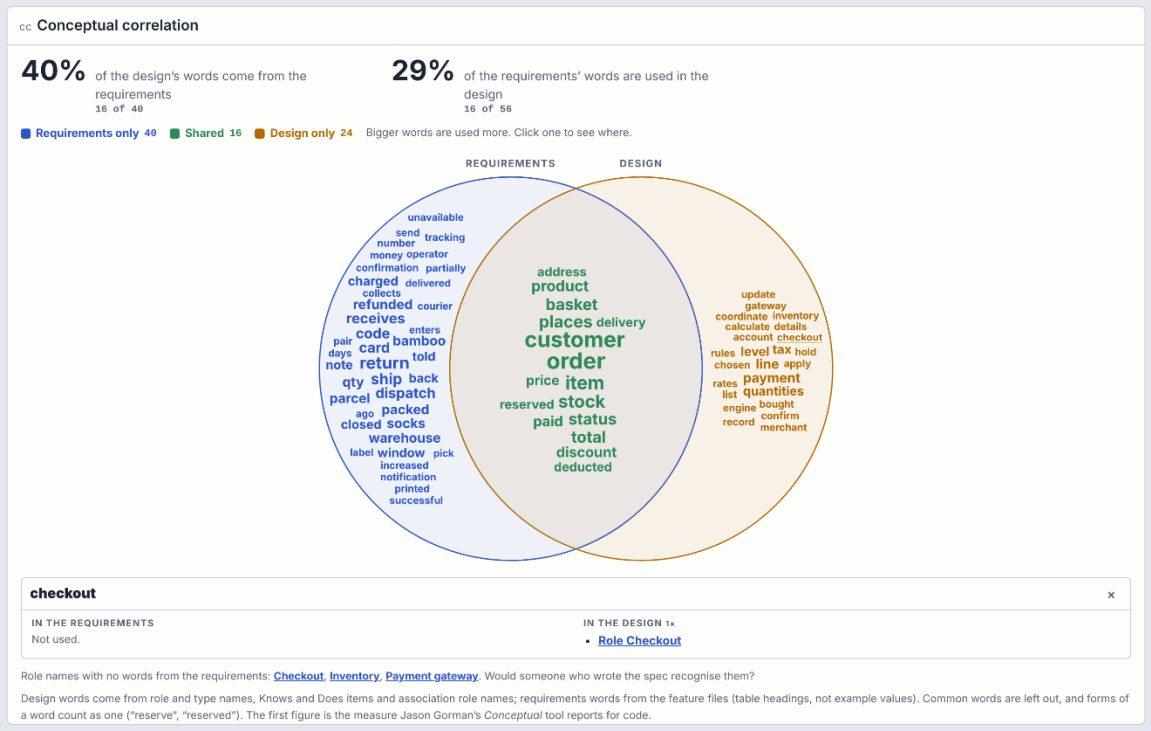
The **Review** tab looks over the whole model for design smells, then compares the design's vocabulary with the spec's. Its findings are prompts for discussion, not verdicts. Cards without a name are reviewed too, shown by their first responsibility: a card is cheapest to reshape before it's named.



Smells. Each finding names the role (or the class, for roles that share a type), the smell, and a sentence explaining what was found. Hover a smell in **Looks for** to see what it means. Set your own limits in **Responsibilities per card** (8 unless you change it) and **Dependencies per role** (4).

Smell	What the workbench looks for
Too many responsibilities	More Knows and Does items than fit on a card
Too many dependencies	Depends on more roles than the limit
Cyclic dependency	Roles that depend on each other, directly or round a loop
Feature envy	A responsibility that uses two or more different features of the same other role
Inappropriate intimacy	Two roles that each use two or more of the other's features
Divergent change?	Different callers use completely separate parts of a role, so it may be doing two jobs. Parts that work on the same information may still belong together. Calls from the Client aren't counted
Data class	Knows things but does nothing with them, or only hands them out ("Get price")
Duplication	The same Knows or Does item on roles of different types
Unused	Not traced to the spec and not called in any scenario

Conceptual correlation is at the bottom of the tab. It asks how closely the design's language matches the requirements', using the measure from Jason Gorman's Conceptual tool for code.



- **The first figure** is the share of the design's words that come from the requirements. It's the one Conceptual reports, so you can compare the design with the code later.
- **The second figure** is the share of the requirements' words that are used in the design. A low figure means concepts the spec talks about that no card covers yet.
- **The diagram** puts the requirements' words on the left and the design's on the right; the shared words sit in the overlap. Bigger words are used more. Click a word to see every spec line and card that uses it; card entries link to the card.
- **Role names with no words from the requirements** are listed under the diagram. Ask whether someone who wrote the spec would recognise them.

Design words come from role and type names, Knows and Does items, and role names on associations. Requirements words come from the feature files: feature and scenario names, narratives, steps and table headings, but not example values. Common words are left out, and forms of a word count as one ("reserve", "reserved", "reserves"), but different words don't ("confirm" and "confirmation").

Saving and sharing your model

Your work saves automatically in the browser you're using. It won't follow you to another browser or computer, and clearing your browser data removes it, so download a copy at the end of each session. Click **Model…** at the top right:

- **Download .json** saves the whole model as a file, named after your first feature and today's date.
- **Copy** puts the model on your clipboard, to paste into a message or another browser.
- **Open .json file…** picks up a model you downloaded earlier, or one someone has sent you. The workbench shows what's in the file and asks before replacing your current model.
- **Load this JSON** replaces your current model with JSON pasted into the box.
- **Load example** and **Start blank** also replace your current model.

Each button that replaces your model asks you to click again to confirm. Models saved from the earlier, roles-first workbench open here unchanged. Models saved here also record what each responsibility needs to know; the earlier workbench ignores that part.

Quick reference

Marker	Where	Meaning
Number on **2 · Name the roles**	Specs	Cards still waiting for a name
Empty / blue / green circle	Specs	Outcome not yet assigned / assigned / realised in its scenario's sequence diagram
Green edge and **knows Price** in Which card?	Specs	That card already knows something the responsibility needs
already does this in Which card?	Specs	That card already has this responsibility
Italic chip with a dashed edge in the Cards tray	Specs	A card with no name yet
Green underline on a word	Specs	Added to what a card knows
Red underline on a word	Specs	Used as a card's name
Dotted red underline	Specs	Mentions a card's name
Blue background on a word	Specs	Mentions the card you clicked in the tray
Diamond by a When step	Specs	A request from the Client; filled once assigned. Not counted in progress
+2 on a chip	Specs	The same responsibility is linked to 2 other outcomes
✓ on a chip	Specs	That assignment is realised in the scenario's sequence diagram
NOT on a chip	Specs	Negative outcome: realised when the responsibility is not called
Orange chip with **Re-add to card**	Specs	The linked item was removed from its card
Name it later and a dashed line under the title	Cards	A card with no name yet
needs … under a Does item	Cards	What that responsibility needs to know
spec / **spec ×3** badge	Cards	Linked to one / three outcomes; hover to see

Marker	Where	Meaning
		which
Filled **spec** badge	Cards	Linked from two or more features
entry badge	Cards	Called by the Client: assigned from a When step
Green / hollow dot by a Does item	Cards	Called / not yet called in a sequence diagram
Dotted collaborator name	Cards	Not used by any call in a sequence diagram
Grey 1 after a collaborator	Cards	Multiplicity not set, so it's 1
Name its role to walk this scenario	Sequence	The outcome's card needs a name before it can take part
Dotted orange message	Sequence	Unreachable: its sender hasn't been called earlier in the scenario
⇠ on a message	Sequence	That call has no return yet; click to add one
↻ on a message	Sequence	The receiver is a collaborator the sender has many of; click to repeat the call
Frame tagged loop	Sequence	The messages inside repeat, under the condition in brackets
Underlined name on a lifeline	Sequence	The role taking part (": Person" adds its type)
Grey operation	Class diagram	Not called in any sequence diagram yet
Number at an arrow's end	Class diagram	Multiplicity: 1 unless set
Orange line under the diagram	Class diagram	Cards left out until they're named
Blue / green / orange words	Review	Requirements only / shared / design only

Common problems

- **I can't add a card to a sequence diagram.** It hasn't been named yet. Name it in **2 · Name the roles**, or click its title on the CRC cards tab.
- **A card is missing from the class diagram.** Same reason: only named cards are drawn. The line under the diagram links to the ones waiting.
- **Several cards look like the same role.** Merge their items onto one card with ↴, then delete the empty card.
- **An assignment is flagged after I edited a step.** Links follow the step's exact wording. Remove the flagged link and assign the step again.

- **A message label turns orange.** The receiving role doesn't have that responsibility any more. Pick one of its responsibilities from the list, or fix the card.
- **A collaborator appears that I didn't intend.** A call that asks for a responsibility makes the receiver a collaborator of the sender, so check the call's sender. Once you correct it, the workbench asks whether to delete the collaboration nothing uses any more.
- **There's no ↻ to repeat a call.** Set the collaborator's multiplicity on the sender's card to * or 1..*.
- **My work has gone.** It was saved in a different browser, or browser data was cleared. Load your last downloaded copy through **Model…**.

© 2026 Codemanship Ltd.