User Journey Map

Complexity Analysis

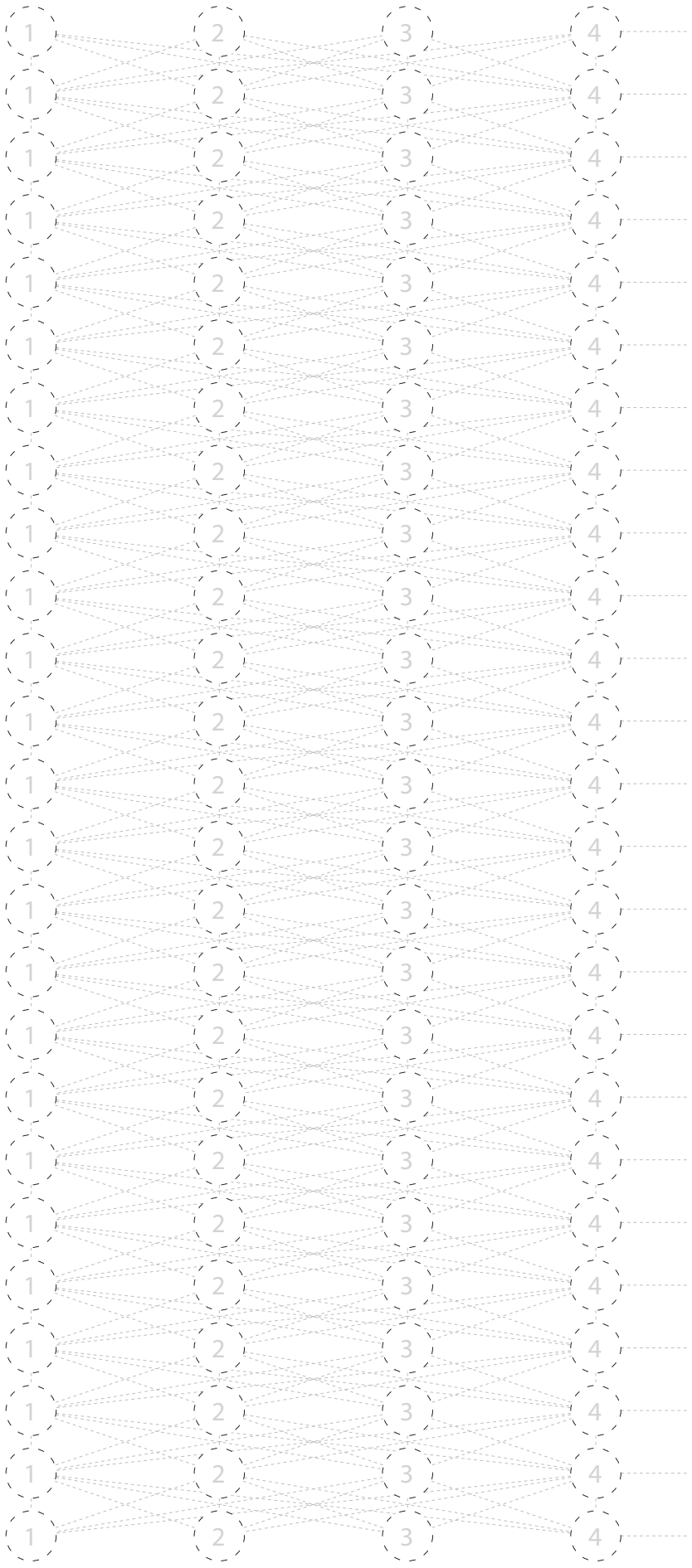
USER GOAL

USER ACTION

SYSTEM RESPONSE

USER DECISION

INTERRUPTION



COMPLEXITY RATING

user-journeys.com

A guide to mapping by
Complexity

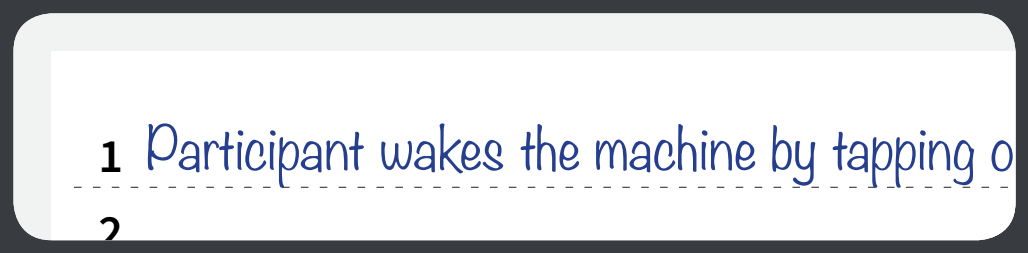
The Complexity Analysis map is designed to help you take an objective view of your product, as you step through the interactions for each task your users attempt to complete with it. As with mapping user journeys against emotion, the idea is to gain empathy with your user. With complexity analysis the task is also to identify unnecessarily onerous interactions. From here you can work to make efficiency improvements to your product, and ultimately help your user complete their goal with your product, quicker.

1. User goal

Start by filling in the name of the goal at the top of the page. This example lists the steps involved with collecting a pre-paid train ticket from a self-service machine at a UK rail station.

2. Step entry

The left-hand side of the grid is where you'll describe each of the steps - in order - that a user would take in order to achieve the goal.

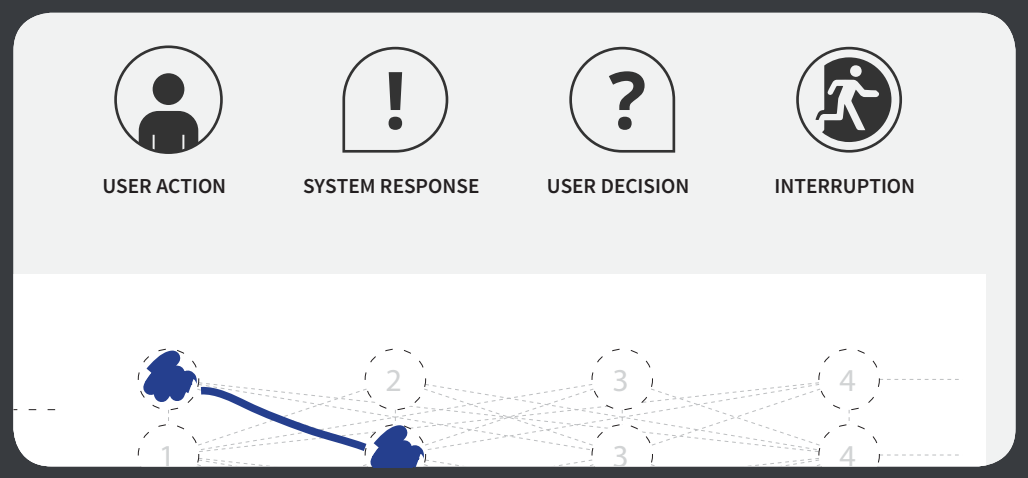


3. Step categorisation

Each step is assigned a type; A user action, a system response, a user decision, or an interruption.

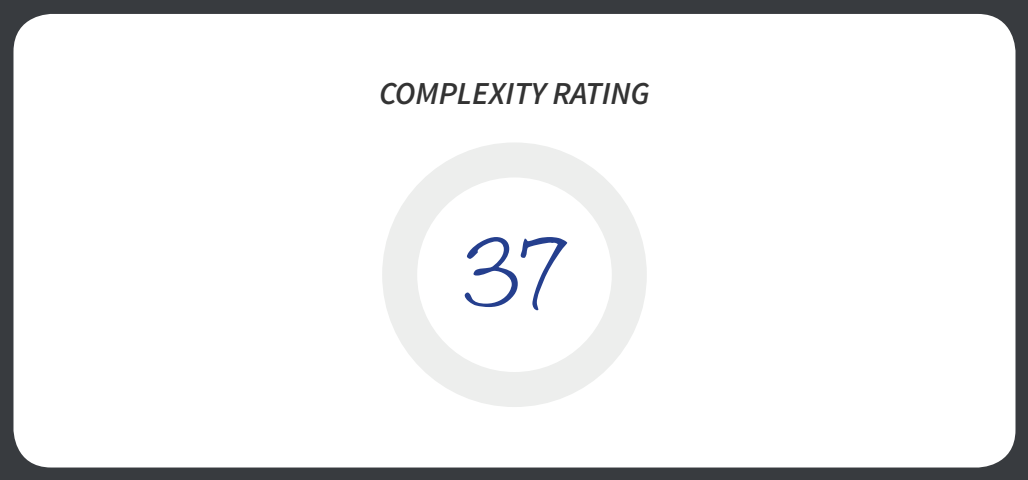
When the user attempts to do something, it's a User Action. A System Response is how your product responds to a user action. When your product presents more than one option, a User Decision has to be made. If the user has to do anything, at any point, that requires their attention be diverted from your product, then that's considered a context Interruption..

Categorise each step by filling in the circle under the corresponding step type. Draw a line to connect the steps together, except when the step type is an interruption. In this case, draw a line off to the right of the page.



4. Calculating the complexity rating

Each step type has a value associated with it (the value of a step type is shown in the corresponding circle on the template). Total up the weighting of each step, and enter it into the complexity rating box.



5. Thinking about error cases

The example above is an overly simplified one (a "red route"). As you use the method you'll most likely encounter the need to account for error conditions (also known as edge cases). Use the same method for exploring user journeys through error states, remembering to detail each of the steps required for your user to leave the error state and complete their goal.