Look at the docs (linked above) and take some notes. What problems do you think this will solve for developers?

This feature will help the developers to design a more customized response for the user. Using this feature, they can add more layers inside one action and thus build an assistant that can give a more in-depth/accurate answer to the user. The number of stories that a developer needs to write in a domain.yml file would be lesser using CRVs. This is also very helpful for developers to build a more contextually aware assistant.

What sorts of questions do you think they'd have?

- How to debug CRV?
- Is there any way where I can write/join multiple conditions in a single block?
- Does it support only boolean types of examples or more advanced types can be used as well?
- Which parameters are optional and which ones are mandatory?
- What is the default value taken by - name, type, value?
- What happens when no condition is matched?
- How does the assistant handle unexpected values for parameters? (Exception handling)

Can you come up with some examples where it would be useful?

1. Rasa assistant is employed by one of the food delivery companies. The user can ask for information on a number of options like - order cancellation, refund process, refund status. Now for each of the options, it will become easier to define a path where the assistant can ask/give meaningful information.
2. A Rasa assistant has been created to keep a track of paid bills. The customer wants to check if they have paid the bill using the assistant but there are multiple services like - internet bill, mobile phone bill, cable TV bill and so on. CRVs can be employed here to answer their questions in a more intelligent way.

What questions or feedback would you have for the product teams?

First, I'll analyze the questions/comments received from multiple sources posted by the developers as well as non-developers. I'd word it as follows :

"This feature seems to be quite helpful for developers based on the feedback we've received. Seems like one of the core asks from the developers is to add debugging capabilities to have an insight into the mind of the assistant so to say. Moving forward, we can focus on adding debugging capability of this feature along with its documentation in the Rasa docs. We can also add a couple of more examples in the documentation as to how "value" can be more advanced data types like string or integer and not just boolean. "

Next, how would you let developers know about this feature? What channels would you use?

There are a couple of ways to promote a newly-released feature within the developer community:

1. Writing a blog post about it on the Rasa's website/youtube video/newsletter.
2. Prepare a use case and publish the tutorial on medium under famous publications like Towards Data Science etc.
3. Submit a post on websites like dev.to, freecodecamp, huggingface.
4. Go on a podcast and talk about how this feature can help with various problems that developers deal with and how this feature can help solve that.
5. Share and publicize the feature on various social media channels like Twitter, Reddit, LinkedIn etc.
6. Maybe give a short-and-sweet talk at local tech meetups or conferences like PyCon about recent features, including this new one.

How would you know if you were successful?

There are multiple ways to measure success in this case :

1. The likes, comments, shares, engagement can be one of the ways to measure success based on the blog posts, videos and articles.
2. The questions from developers or users opening a pull request, issues and increased engagement with Rasa's github repository will help to measure the success amongst the developer community as well.
3. Maybe we can have a poll on Rasa's website on the feature's documentation page asking developers if they found this useful or not (a feedback poll of sorts). This can be a temporary poll if needed (for the first X numbers of days maybe).

3. Create a short piece of content introducing the concept. Some examples:

1. A slide

2. A forum or blog post

3. A short video

Here is the link for a blog post - https://github.com/kajal-puri/article-rasa/blob/main/crv.md