HOW TO WRITE GREAT USER STORIES

What is a User Story?

"User story is an informal, natural language description of one or more features of a software system."

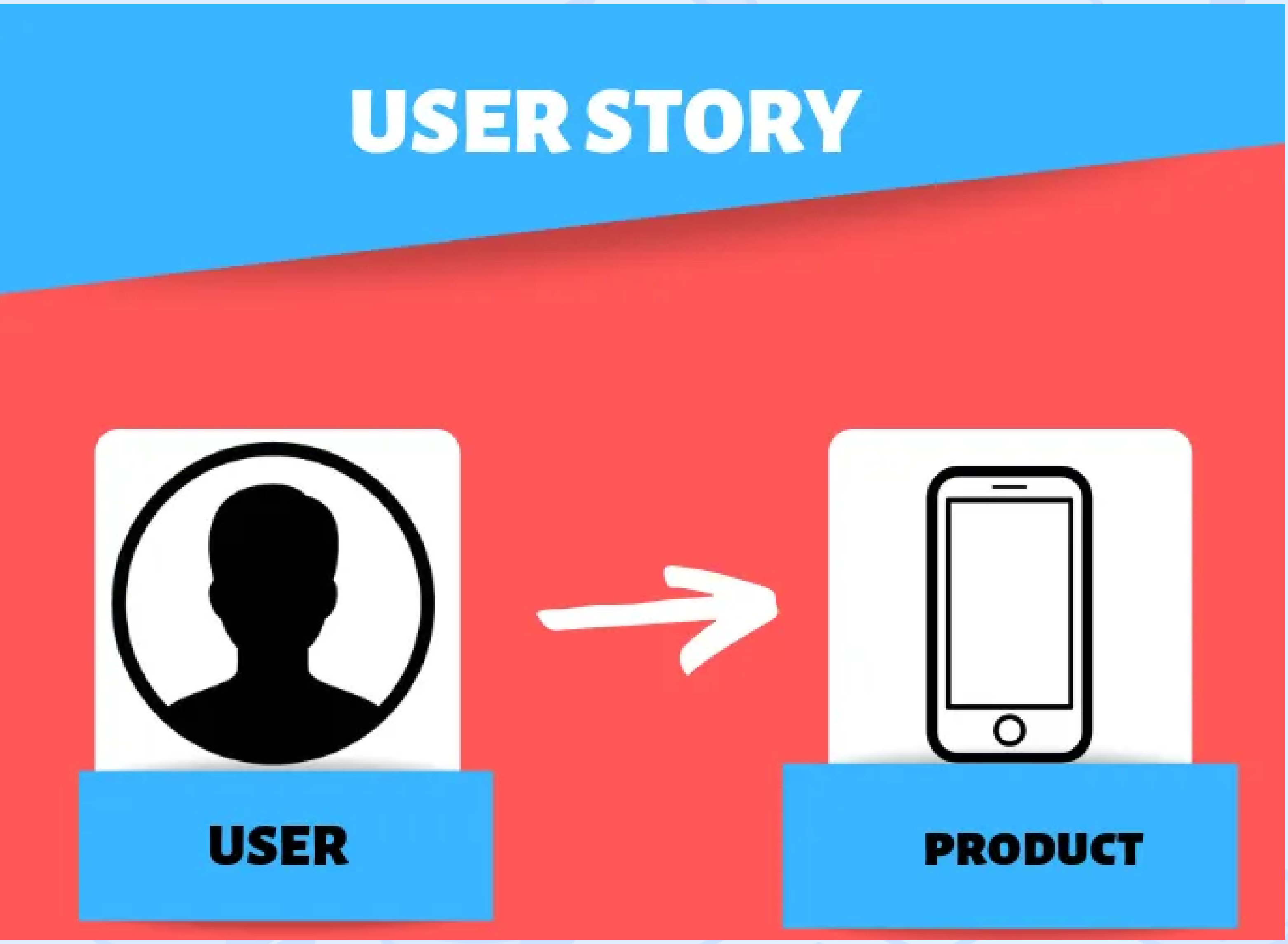
In modern software development, user stories are one of many steps to a successful implementation of the Agile approach. If you aren't yet familiar with Agile, read this article beforehand.

In Agile, there are many different terms and principles that need clarification – so today we want to share some insights about what user stories are and how to successfully write them.

- **Agile user stories are short descriptions of user needs that explain why you need to make a particular feature for your digital project.**

Now let's dive into details.

The structure of a User Story



Their structure is:

As a "Who", I want "What" so that "Why".

"Who" here means a person interacting with the product, **"what"** means an action or a process of interaction and **"why"** means the end goal of that interaction (why it should happen in the first place).

For example:

"As a Sports Fan, I want to view the score of the latest match so that I knew if my team won".

That's it! Although user stories may look primitive, especially to a seasoned business analyst who's used to describing everything in detail, their simple structure is the reason they're so popular and widely-used in Agile software development.

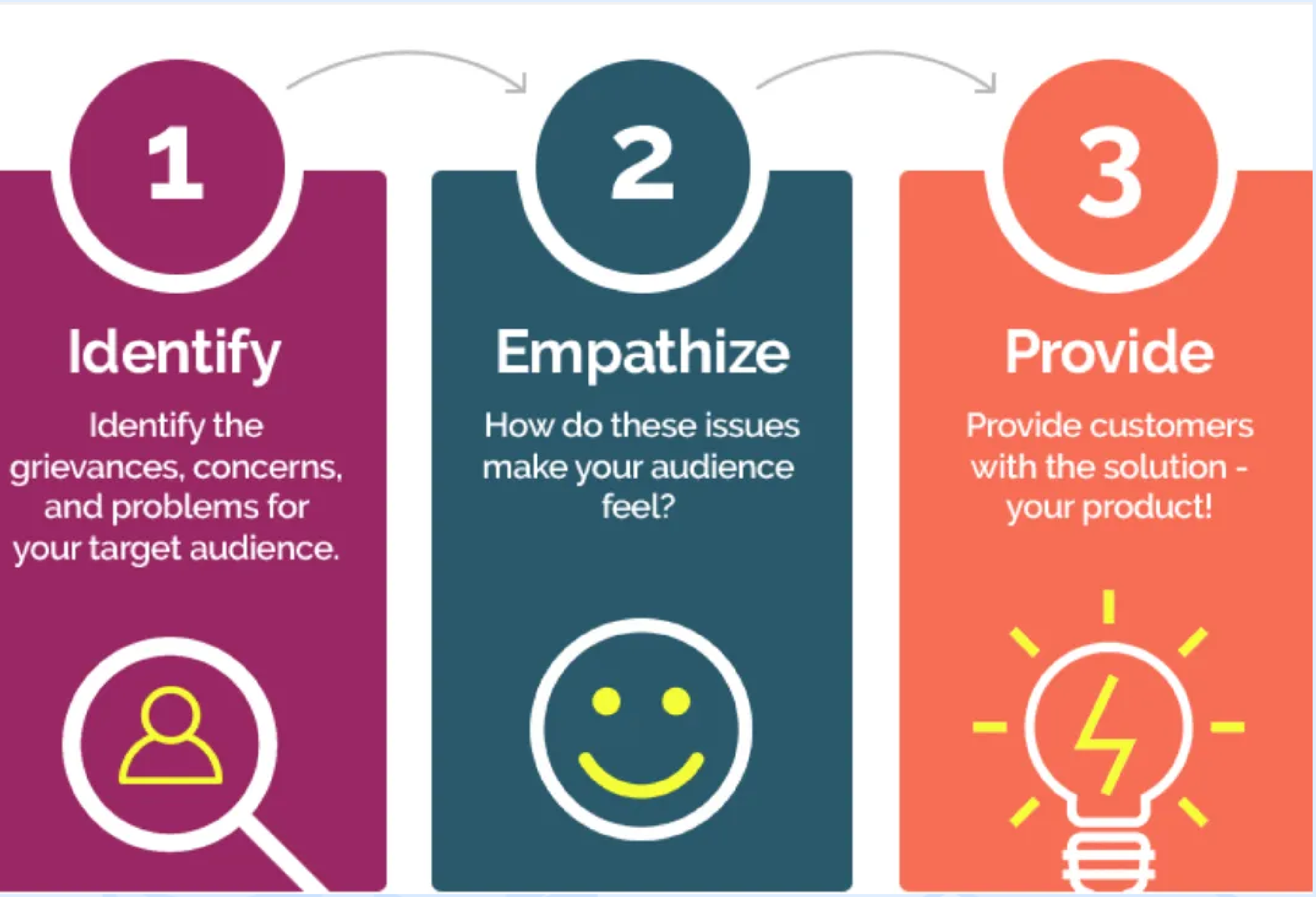
NB: User stories are usually written by product owners. This is one of the roles in this particular software development approach as described here. The Product Owner can be from the developer's side or the client's side, or can be a third-party.

They have a vision of the project, collect all the ideas into a wish-list (backlog) and translate it to the developer's team. One of the ways of effectively explaining the necessities of the future product is to describe it in the format of User Stories.

The idea is that you as a product owner will be able to look at a user story, quickly recall a feature and be able to explain it to developers or stakeholders.

So how do I write a genuinely useful user story for task-setting during development? Though dozens of books have been written about applying user stories, it's not that difficult to start creating the stories for your project just by following the guideline below.

Step-by-step instructions for creating working User Stories


1
Identify
Identify the grievances, concerns, and problems for your target audience.
2
Empathize
How do these issues make your audience feel?
3
Provide
Provide customers with the solution - your product!

First: Find out who your user is

1) Start with the User

A distinctive principle of user stories is "people go first".

A simple structure saves you time on going into unnecessary details.

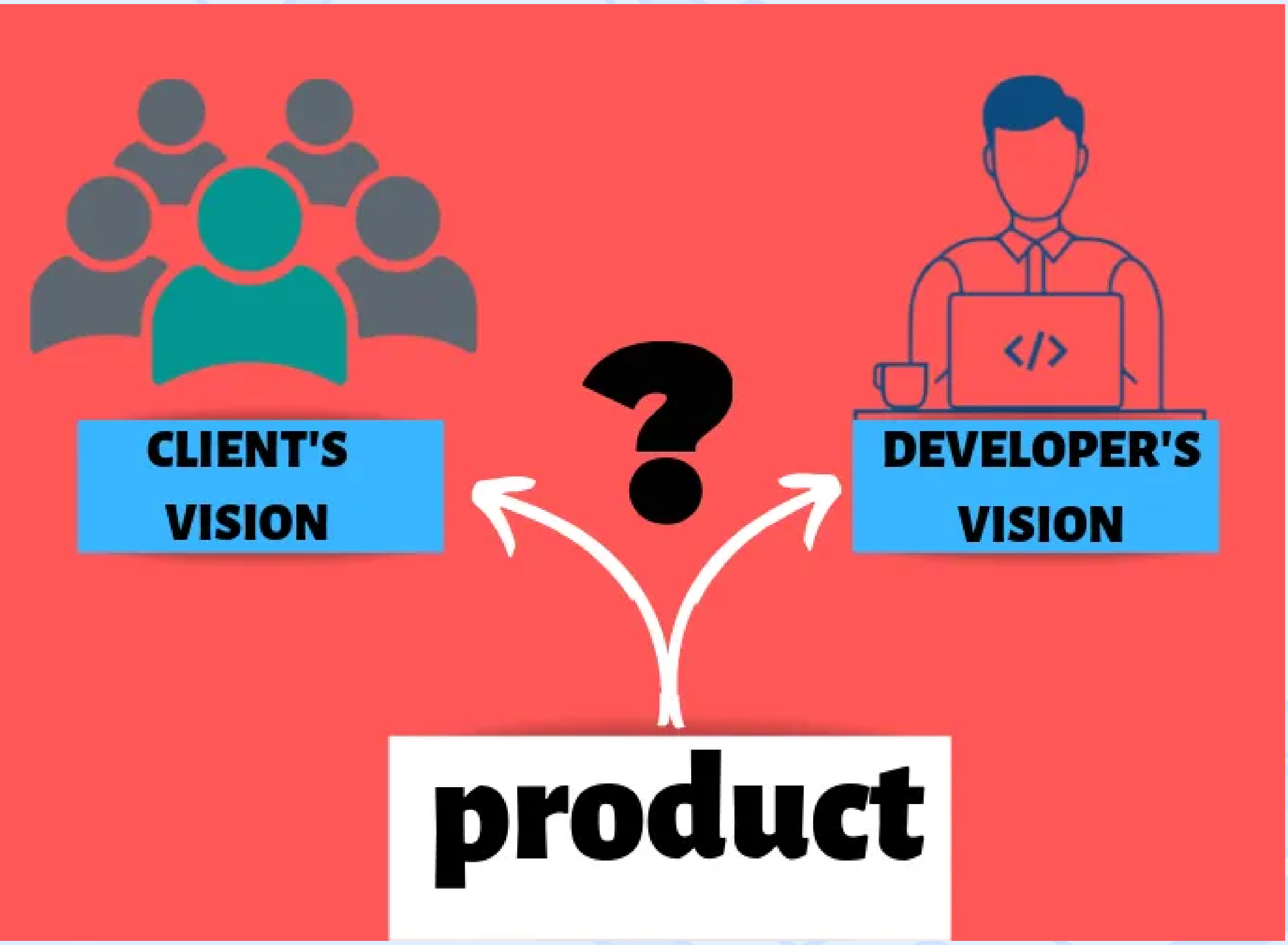
- Think of who’s going to use your product, as well as who stands to benefit from it. Often the first person to come to mind isn’t the right one. For example, it isn’t usually the end user that benefits from signing up but the business owner. By adding a sign-up layer the owner ensures the app is working in accordance with privacy policy regulations, as well as getting the chance to collect user data (with their permission, of course).

2) Exclude everything that doesn't help

Dismiss from the feature any actions which add no value.

- Think of what your end user really needs. Do not develop anything that looks attractive but does not solve the user problem. This is important to remember when you proceed to the next step.

Then: Discover your target feature



1) Think of how you can solve the problem

The "What" is a way to briefly explain how the functionality you're building will actually work.

2) Try to use widely recognisable notions

For example, "catalogue". Everyone imagines more or less the same thing when they hear "catalogue": a list of items with images/names/sometimes descriptions.

- Try this yourself. Call "a tool that allows for the selection of multiple items during the process of purchase" a "shopping cart". But don't use industry-specific words unless absolutely necessary. You don't want to explain the concept of a Chiropractor to developers every time, do you?

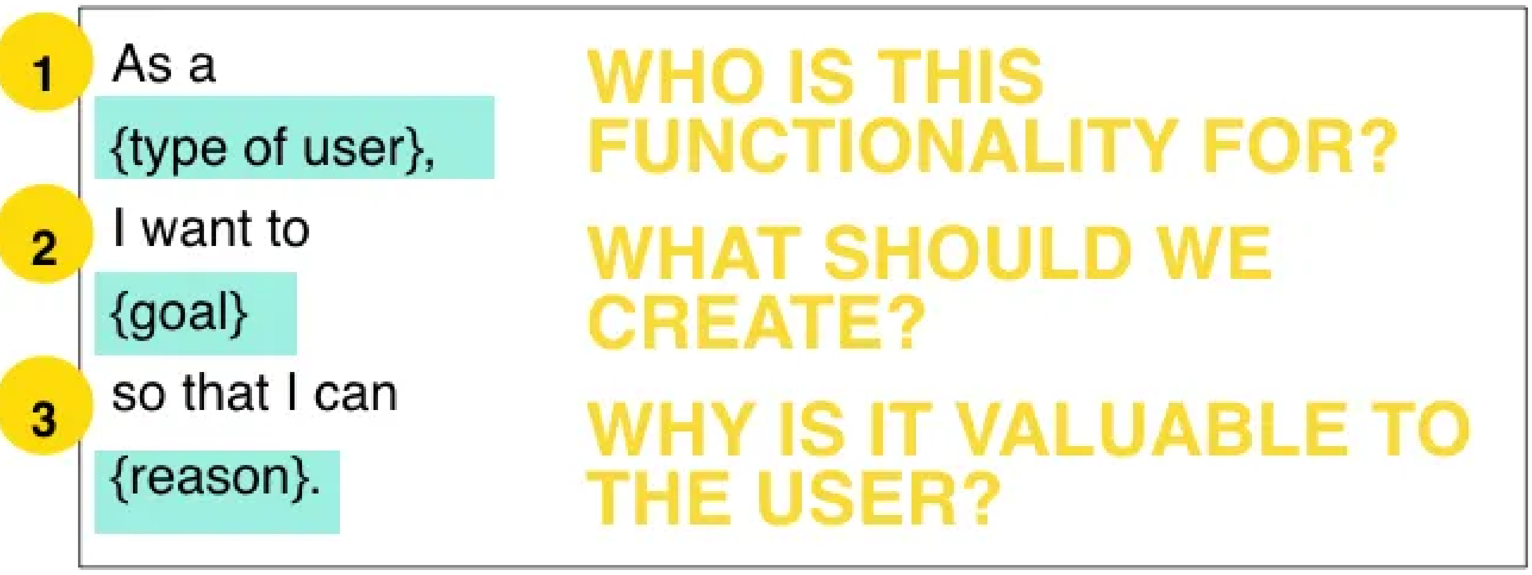
3) Don't overthink it

User stories are not a stand-alone tool. They are meant to help trigger a conversation with everybody, describing what you mean in detail.

- Think of them as of internal memos: every one should get a general idea, but you don't have to squeeze the overall complexity of a solution into two or three-word "what".

Your stories aren't going to participate in a "Best User Story" competition. Just make sure you've explained your idea loud and clear and that the developers get it.

And finally: The "Why"



1) Write down the reason behind the task

The third step is where you finally capture the purpose of building the feature you're working on .

- Don't limit yourself here – your product doesn't necessarily have to be life-defining for the user. But this section helps you prioritise which parts of your product are solving real, big problems and which are "nice-to-have".

To compare with the "Football Fan" example above, here's an example with the same User but a different "Why":

As a Football Fan, I want to blow a Vuvuzela so I can support my team.

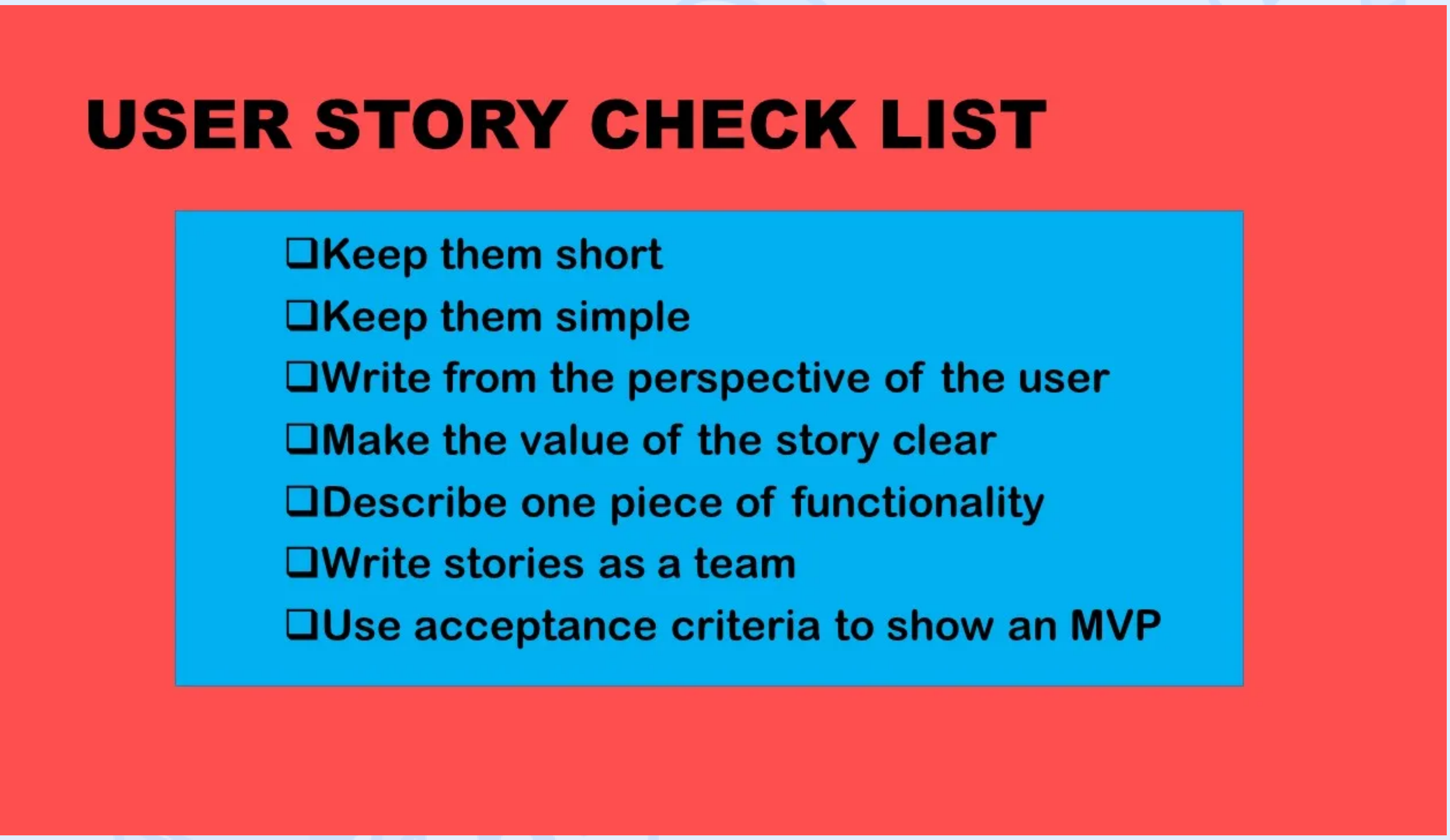
This one also has a "Why", but... Can you see the difference?

2) Make sure the solution meets the problem

Does a Vuvuzela meet the need to support my favourite football team? Perhaps. Maybe there are other ways to support them, like cheering or buying merch?

- There’s more than one way to solve every problem, but the "What" always stems from the "Why". Understanding the real needs of the User is the key to meeting their needs.

That's everything you need to write productive user stories. Also remember, no matter how professional you as a product owner are, always show your stories to your team and discuss them together. Improve and rewrite them with the help of your colleagues to ensure the highest possible quality. You won’t be able to achieve it alone.



USER STORY CHECK LIST

❑ Keep them short
❑ Keep them simple
❑ Write from the perspective of the user
❑ Make the value of the story clear
❑ Describe one piece of functionality
❑ Write stories as a team
❑ Use acceptance criteria to show an MVP

Final point: add acceptance criteria

Acceptance criteria are a vital component for the assessment of user stories' potential for successful realisation after the coding is done.

- I've added bullets below user stories (like in description to a Trello Card) explaining what the final product should allow the user to do. This way, developers and testers can check that the functionality is working in accordance with the user's expectations.

Coming back to the Football Fan example, these would look something like this.

"As a Sports Fan, I want to view the score of the latest match so that I know if my team won"

Acceptance Criteria:

1. A Sports Fan can check the teams playing in the latest match.
2. A Sports Fan can view the number of points scored by each team.

In case you prefer a more detailed explanation of user stories, watch this video.

How detailed should a User Story be?

A good user story is a well-balanced description, neither too detailed nor unclear. You should say just enough to fully encompass the value that the given feature needs to deliver. It should be clear, concise and objective.

However, if the case is too complicated, there are two ways to handle it:

- ***Divide the story into smaller substories. This will permit you to add more detail.***

For example, if you want to enable the user to share content on social networks, do not write “As a user, I want to share content on social networks”. It’s too vague. Write: “As a user, I want to share content on Facebook…”, “As a user, I want to share content on WhatsApp…” and so on.

- ***Another way is to leave your user story in peace but add a broader description for the acceptance criteria.***

For example, our user story “As a user, I want to share content on social networks” stays the same, but the acceptance criteria are sharing on Facebook and WhatsApp.

How do you choose between the two approaches? If the feature is too big to be realised within one iteration, subdivide it. If not, opt for the second option.

That was our guide on writing user stories for successful apps and software. If you’re willing to learn more about how we work with Agile, stay tuned for new posts.