

# Acceptance criteria checklist for Agile user stories

Acceptance criteria define the conditions that need to be met to complete a user story. This checklist shows how to collaboratively create acceptance criteria that deliver more value, sooner. When you're checking your acceptance criteria, remember this mnemonic:

## INVEST in the 7Cs

Your acceptance criteria should be:

Independent	Avoid acceptance criteria that depend on other stories where possible. Dependencies cause delays.	<input type="checkbox"/>
Negotiable	The product owner drafts, or gets someone to draft, the acceptance criteria. However, these are to prompt a conversation with the developers, not to prescribe exactly how they implement the story. The whole team discuss and agree on the acceptance criteria.	<input type="checkbox"/>
Valuable	Writing the acceptance criteria from the user's point of view helps ensure the result delivers value to the user.	<input type="checkbox"/>
Estimable	The development team must have enough detail to estimate the size of the story in backlog refining and iteration planning. This means the product owner has to make sure there are draft acceptance criteria ready for these discussions.	<input type="checkbox"/>
Small	The work described in the acceptance criteria should be as small as possible. Do the minimum required to deliver value. Smaller stories deliver value faster, so check if some acceptance criteria can be split off into a new story.	<input type="checkbox"/>
Testable	You must be able to answer Yes or No to the question, "Has this acceptance criterion been met?". This means developers can write automated tests and the product owner can easily check the story when it's up for acceptance.	<input type="checkbox"/>

Clear	Use plain language. Remove vague or ambiguous terms. Discuss the acceptance criteria until everyone has a shared understanding. Capture the results of your discussions so you don't forget them. This means the stories will make sense to the team later and to other stakeholders.	<input type="checkbox"/>
Concise	Limit the number of acceptance criteria. The more there are, the more likely the story should be split. Limit the amount of detail. Things like business rules or content can go in attachments to the story. Don't duplicate what's in your Definition of Done.	<input type="checkbox"/>
Consequence-focused	Focus on the consequence not the construction, the outcomes not the implementation. Describe what benefits you'll deliver your customers, don't prescribe how you'll deliver them. This empowers the developers to find the best solution. If your acceptance criteria are a list of tasks, you're focussing on 'how' not 'what'.	<input type="checkbox"/>
Collaborative	Anyone in the team can draft them, and everyone contributes by ensuring that they tick each box in this checklist. Since the product owner is responsible for the backlog items and their acceptance criteria, the product owner needs to be happy with the end result.	<input type="checkbox"/>
Changeable	Just as they can change in refining and planning, acceptance criteria can, if developer and product owner agree, change once development is underway, as new information comes to hand. You want to respond to change rather than stick to the plan.	<input type="checkbox"/>
Constraining	Acceptance criteria define the boundaries of the story. Developers deliver what's needed to meet the criteria and no more, maximising the work <b>not</b> done.	<input type="checkbox"/>
Customised	Experiment to find what works best for your team. Acceptance criteria can get too detailed, granular and prescriptive, or be too vague, broad and open. Find where on the spectrum your goldilocks zone lies.	<input type="checkbox"/>