Practical Agile for Software Product Management

Concepts, Techniques, and Tools to Develop and Manage Market-Fit Software Products

Module 2

Introduction to Scrum Rituals & User Stories

Ali Samanipour

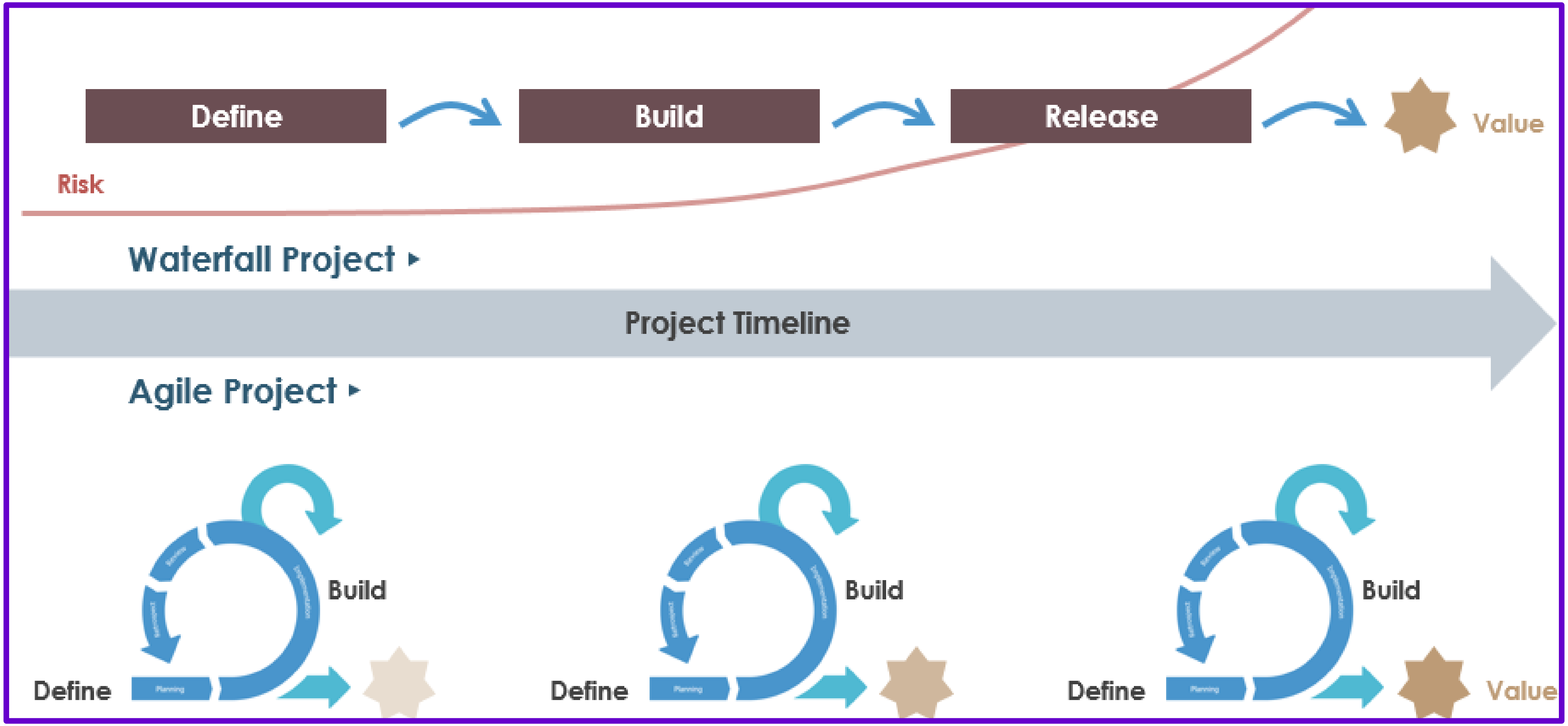
May. 2023

What You Will Learn

→ Scrum Rituals: Daily Standups & Sprint Planning

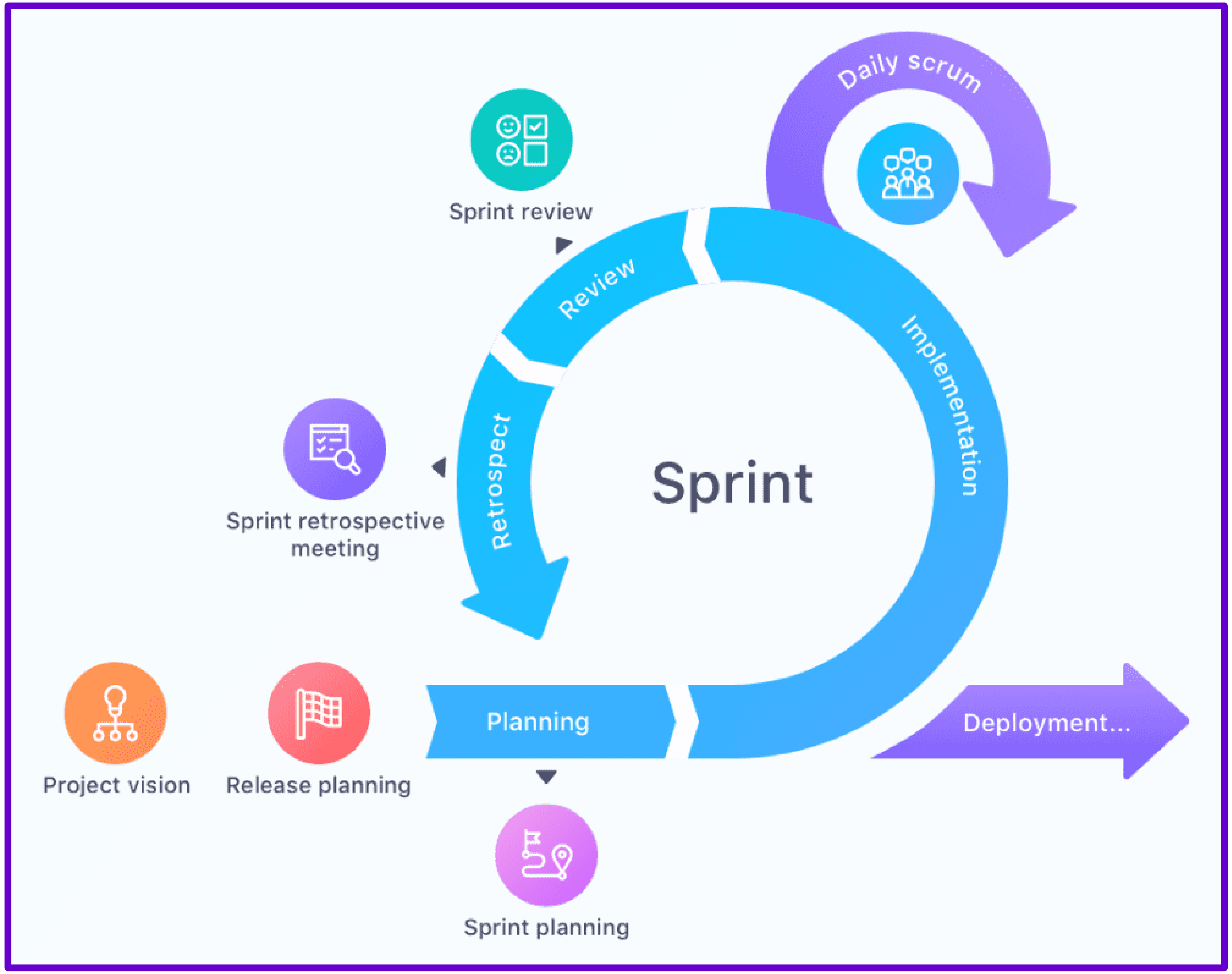
2 Scrum Artifacts: User Stories

Remember: Waterfall vs Agile



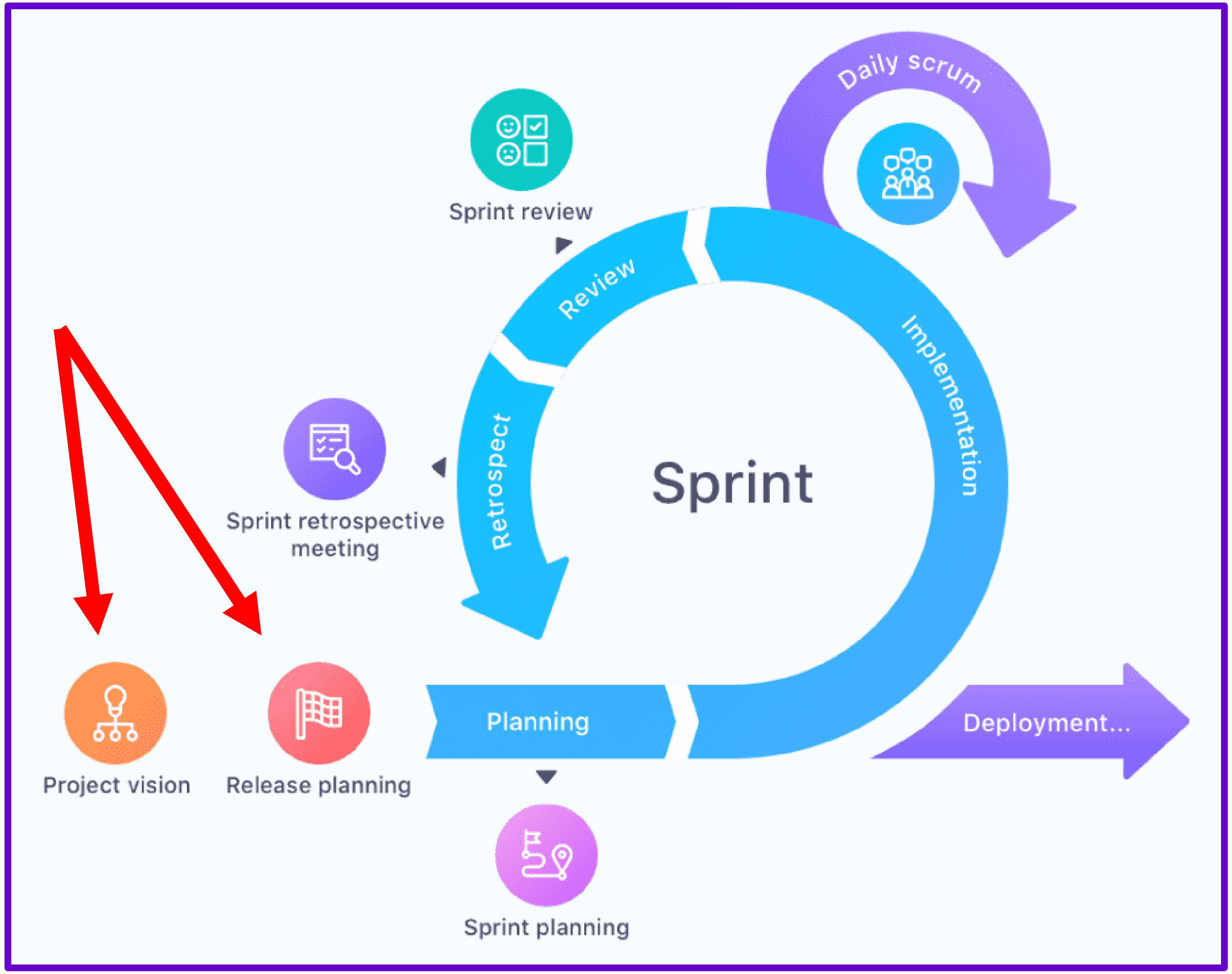
Sprint Rituals

A Time-boxed face-to-face gathering in real time, which offers people the opportunity to have targeted communication with each other about the context of that work.



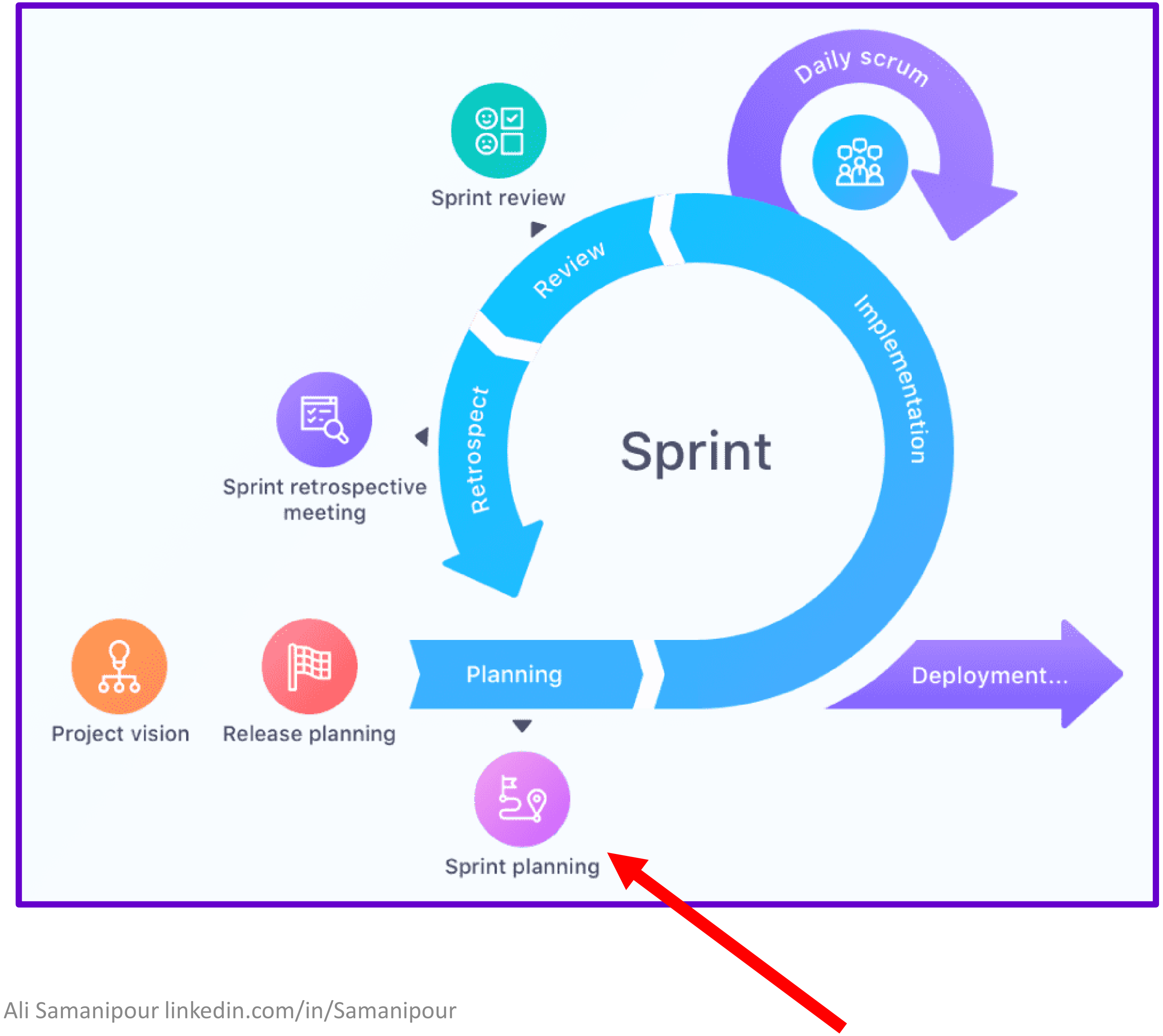
Before Starting a Sprint

Review (Define) your product Vision, Mission and Release plan



Sprint Rituals:
1. Sprint Planning

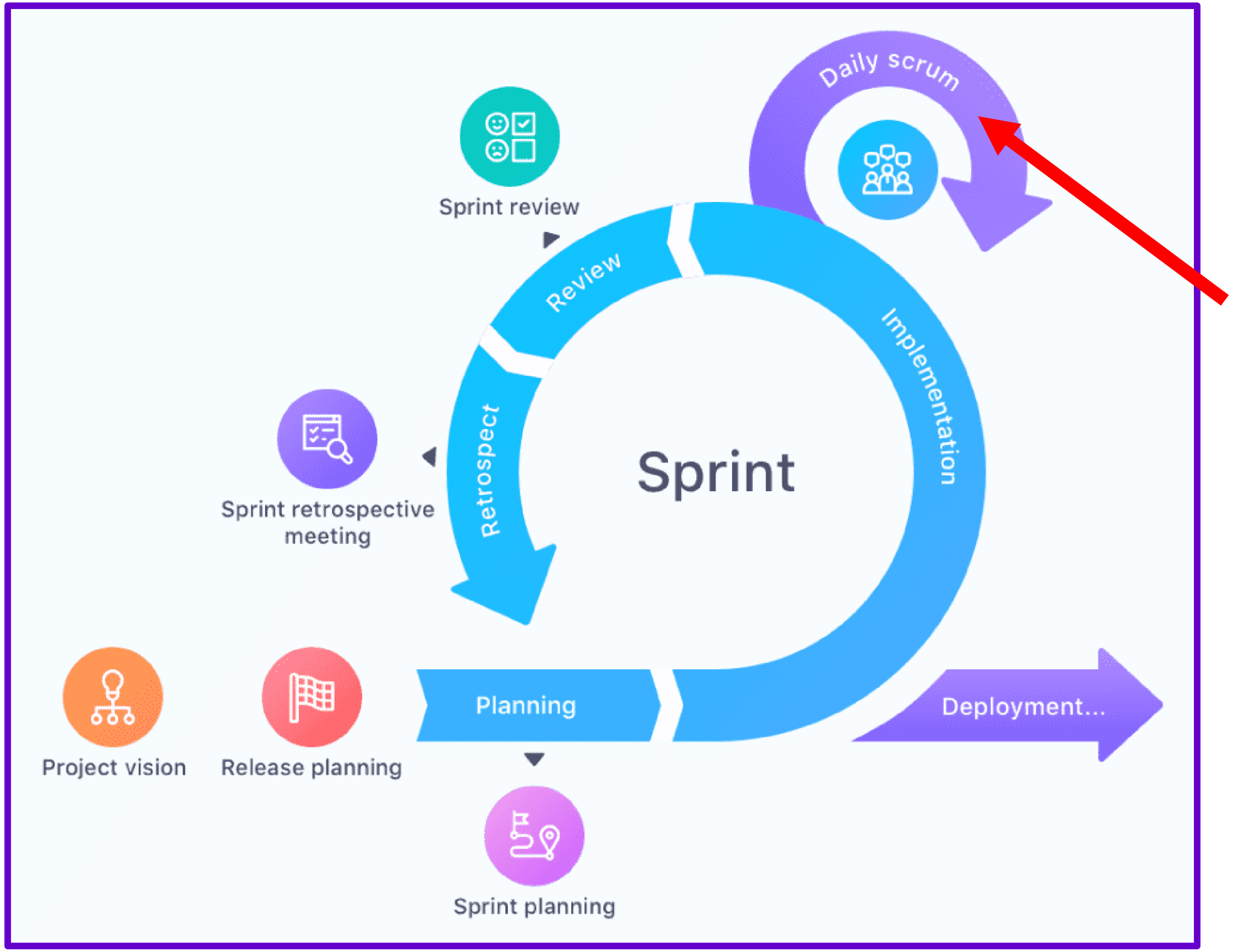
1. Introduce New Stories by PO
2. Estimate Story Points by Dev Team
3. Commit Stories To Backlog



Ali Samanipour linkedin.com/in/Samanipour

Sprint Rituals:
1. Daily Standup

1. What have you done since the last stand up?
2. What do you plan to do until the next standup?
3. Is there anything blocking your progress?



What You Will Learn

✓ Scrum Rituals: Daily Standups & Sprint Planning

→ Scrum Artifacts: User Stories

Scrum Artifacts

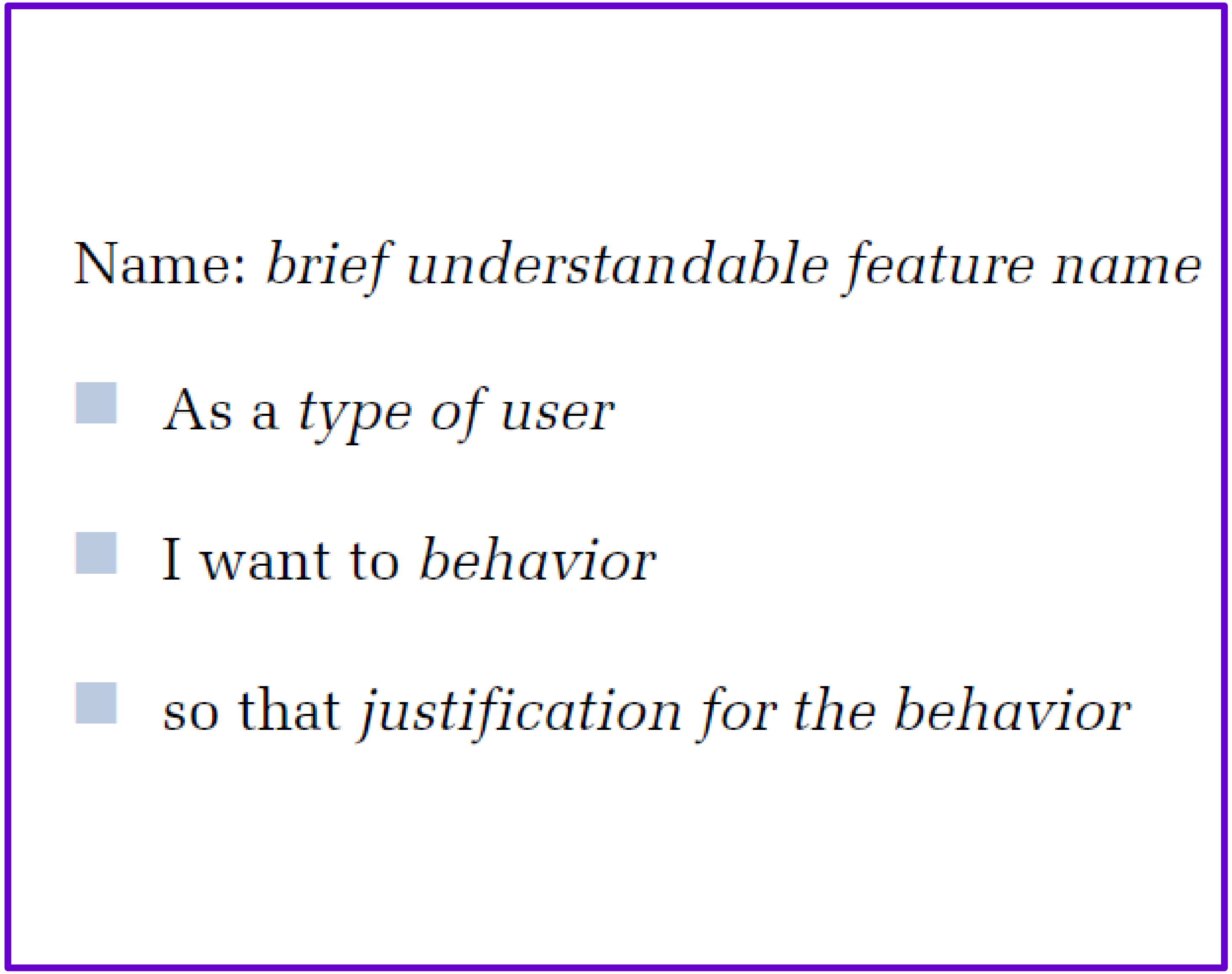
Artifacts are the tools of scrum that allow teams to manage their scrum process.

Anyone can point to the artifacts of a scrum team to show people what they’re working on, what they've accomplished, and what they expect to be able to achieve.

■ story ■ product backlog ■ sprint backlog ■ scrum board ■ definition of "done“■ velocity charts ■ burndown chart ■ product increment

Scrum Artifacts:
Stories Definition

Stories Aren't Technical Specifications

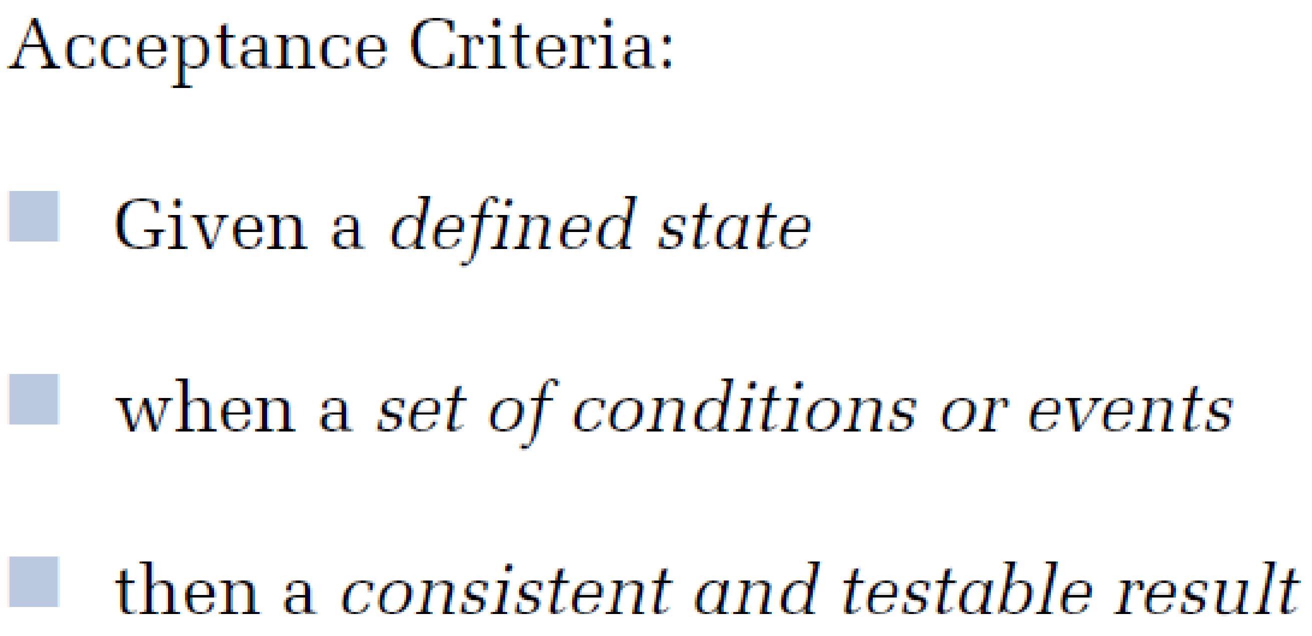


Name: *brief understandable feature name*

- As a *type of user*
- I want to *behavior*
- so that *justification for the behavior*

Scrum Artifacts: Stories Acceptance Criteria's

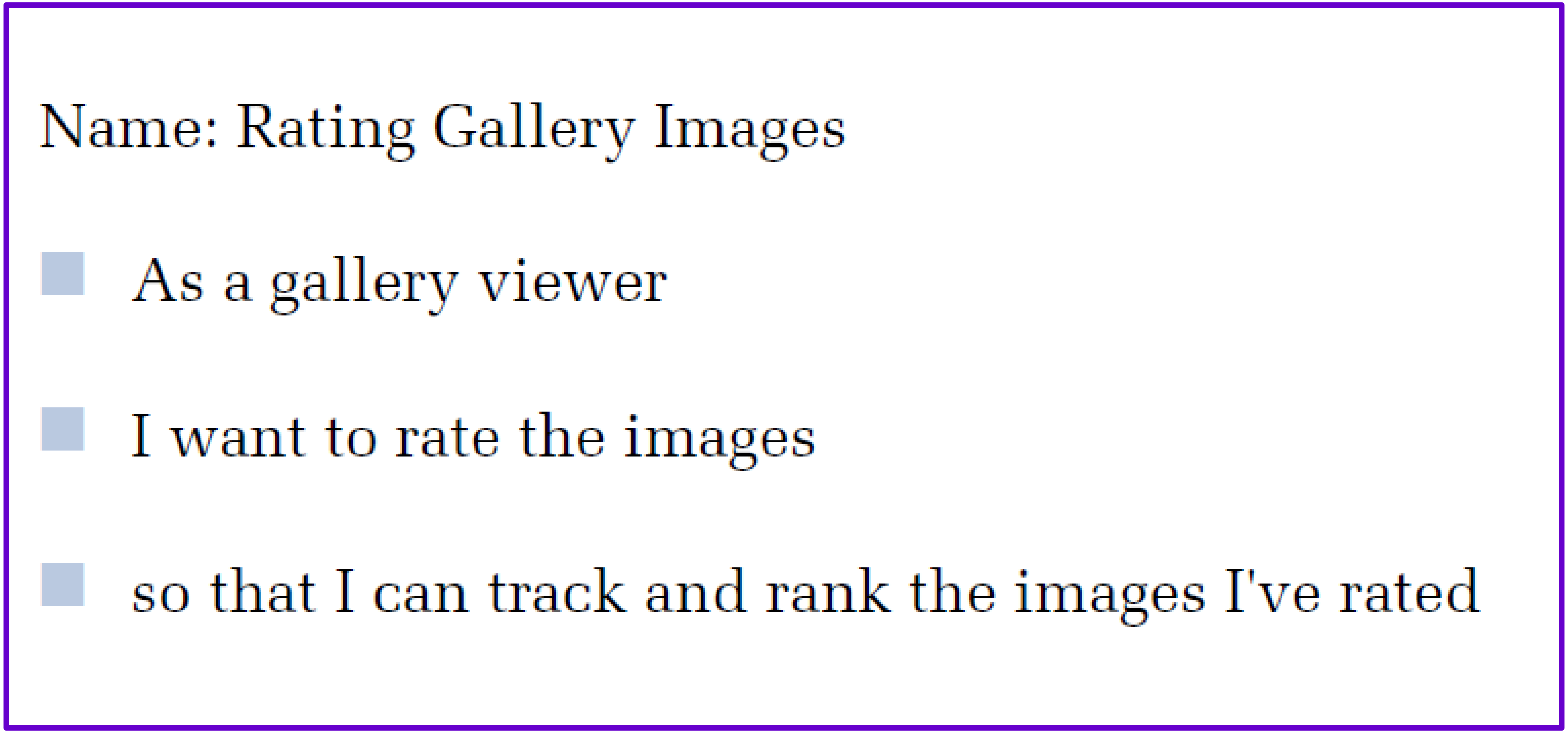
Acceptance Criteria's Are Technical Specifications



Acceptance Criteria:

- Given a *defined state*
- when a *set of conditions or events*
- then a *consistent and testable result*

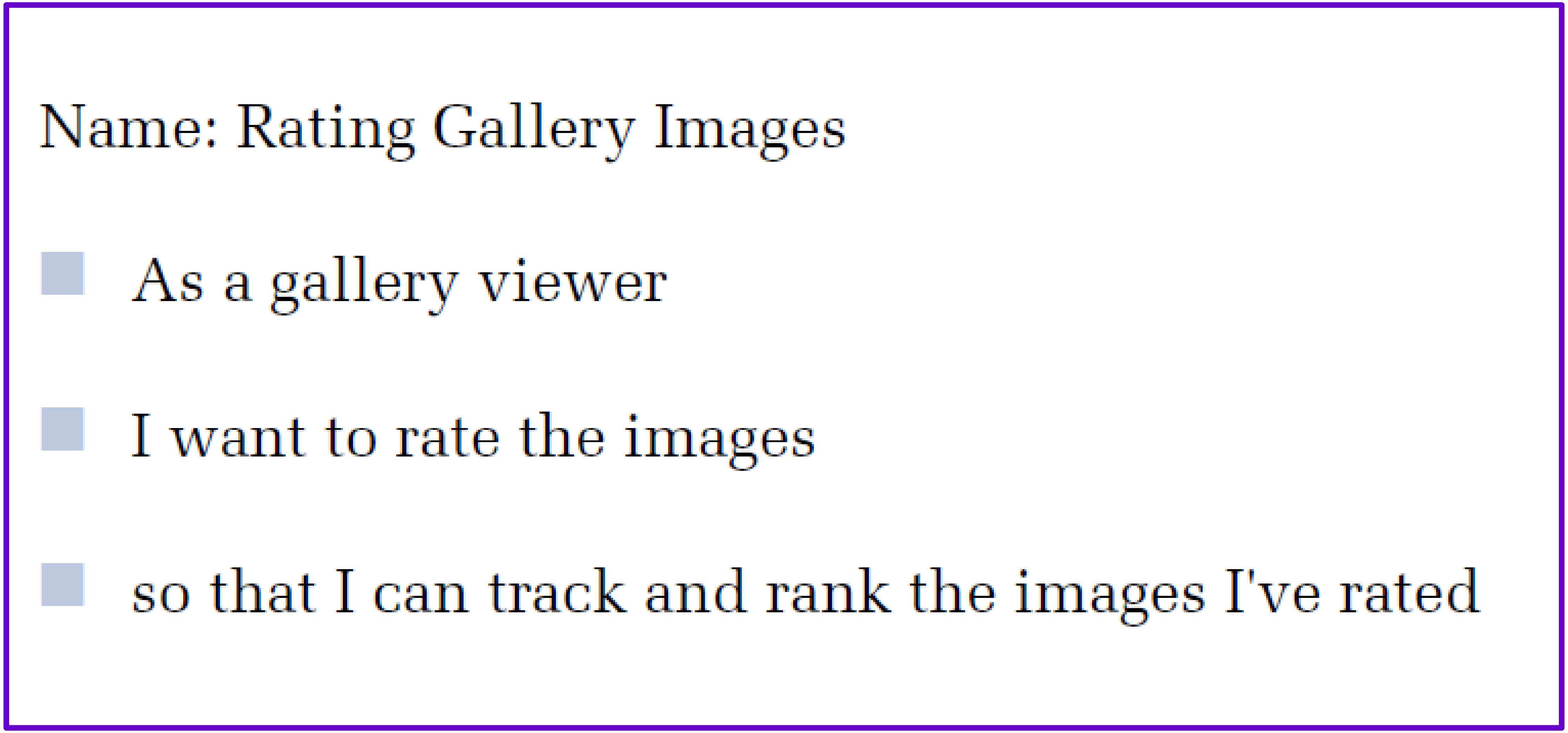
Story Example



Name: Rating Gallery Images

- As a gallery viewer
- I want to rate the images
- so that I can track and rank the images I've rated

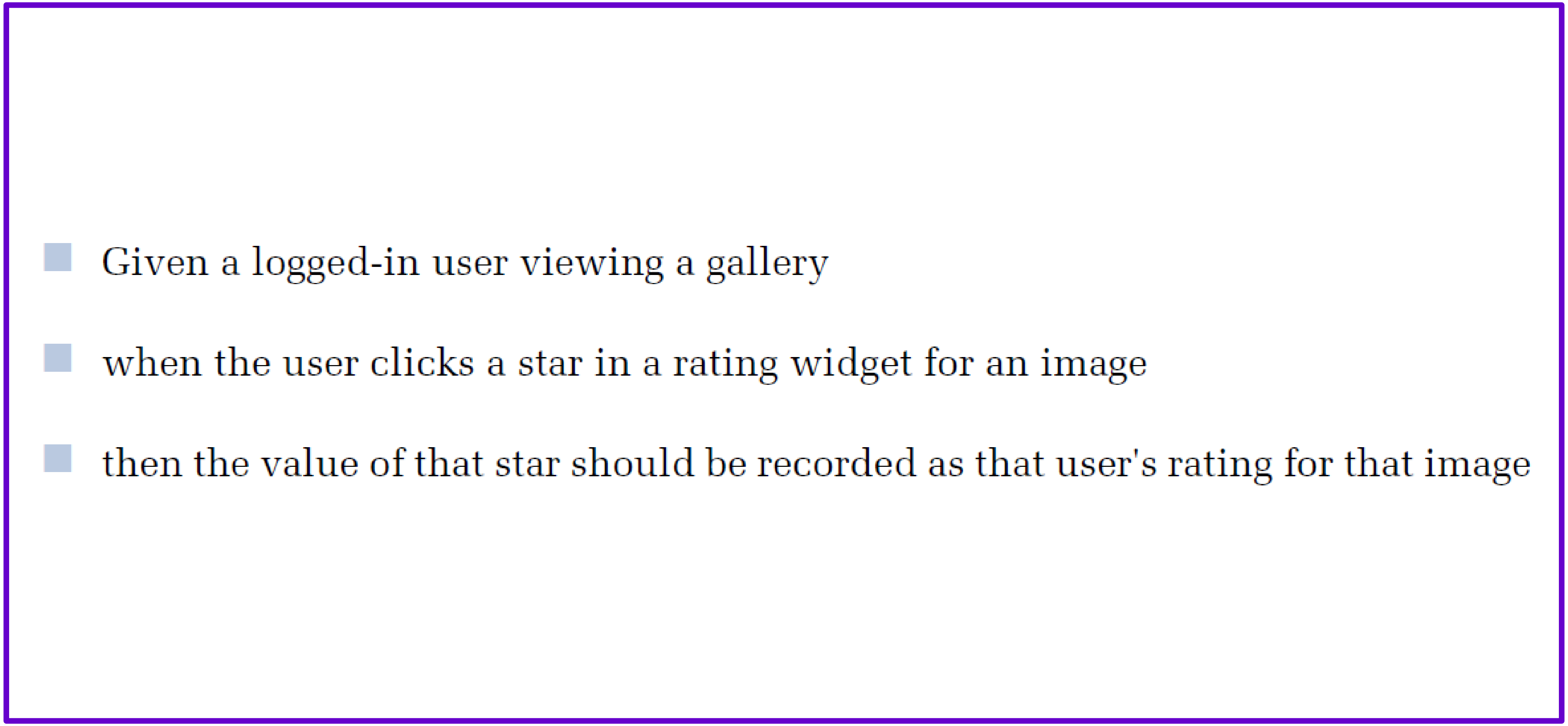
Story Example



Name: Rating Gallery Images

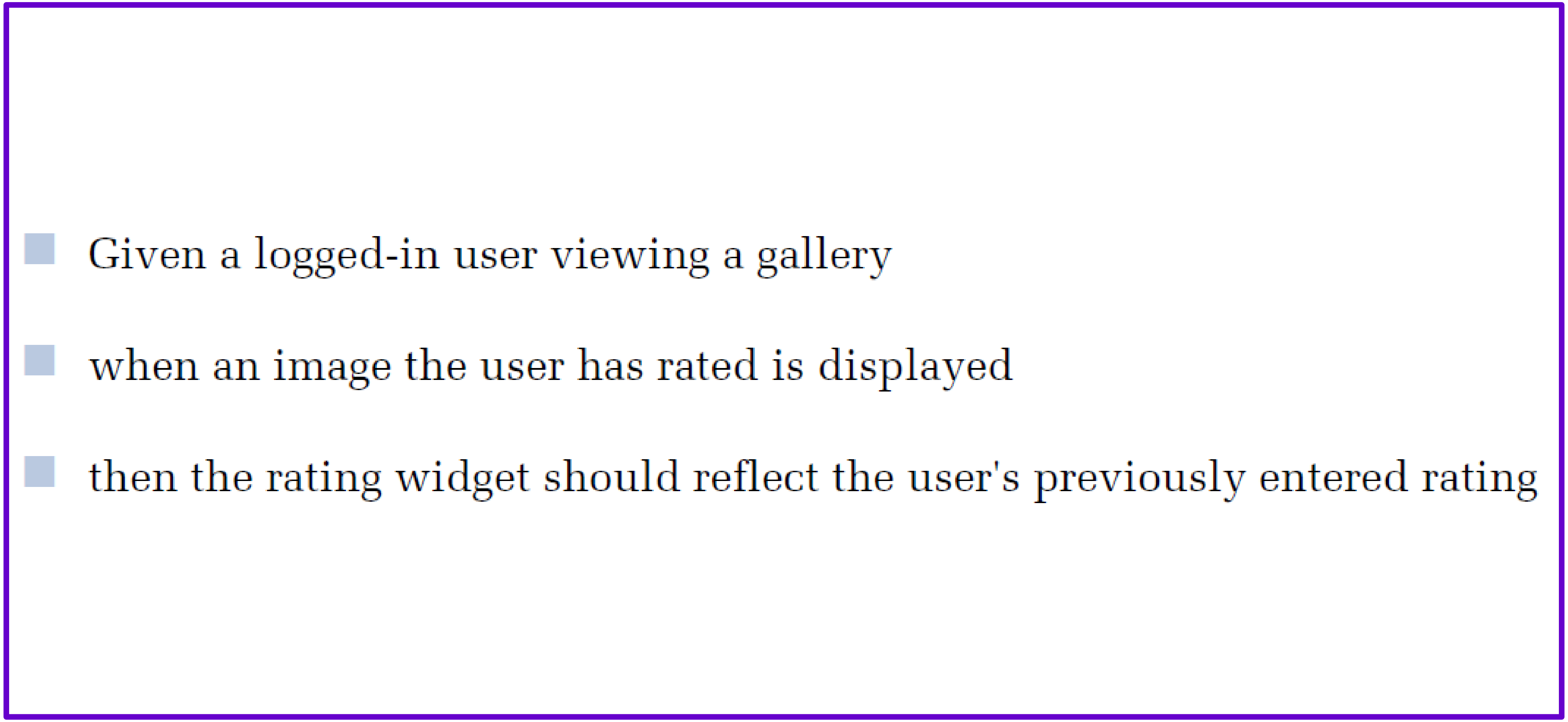
- As a gallery viewer
- I want to rate the images
- so that I can track and rank the images I've rated

Story Example: First Acceptance Criteria



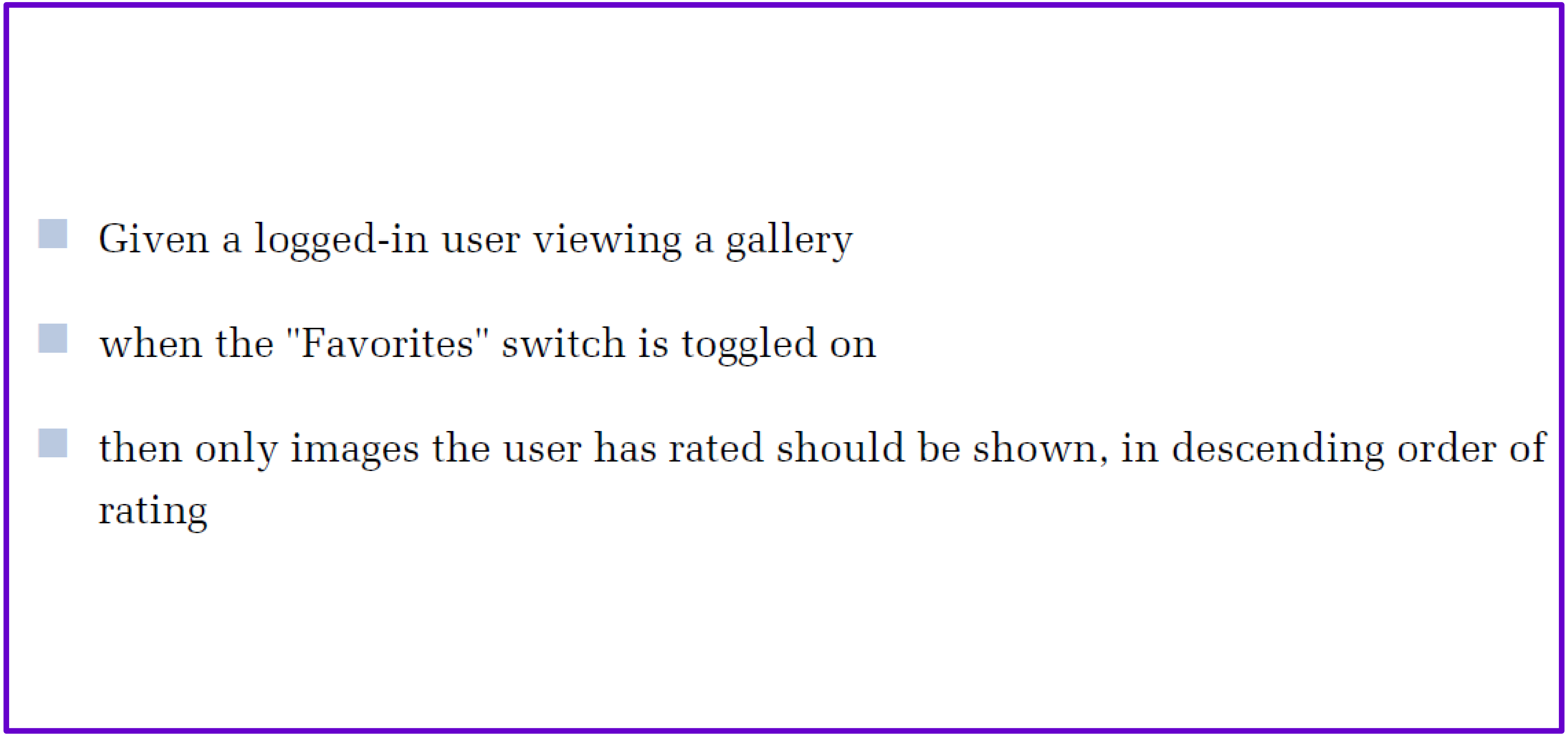
- Given a logged-in user viewing a gallery
- when the user clicks a star in a rating widget for an image
- then the value of that star should be recorded as that user's rating for that image

Story Example: Second Acceptance Criteria



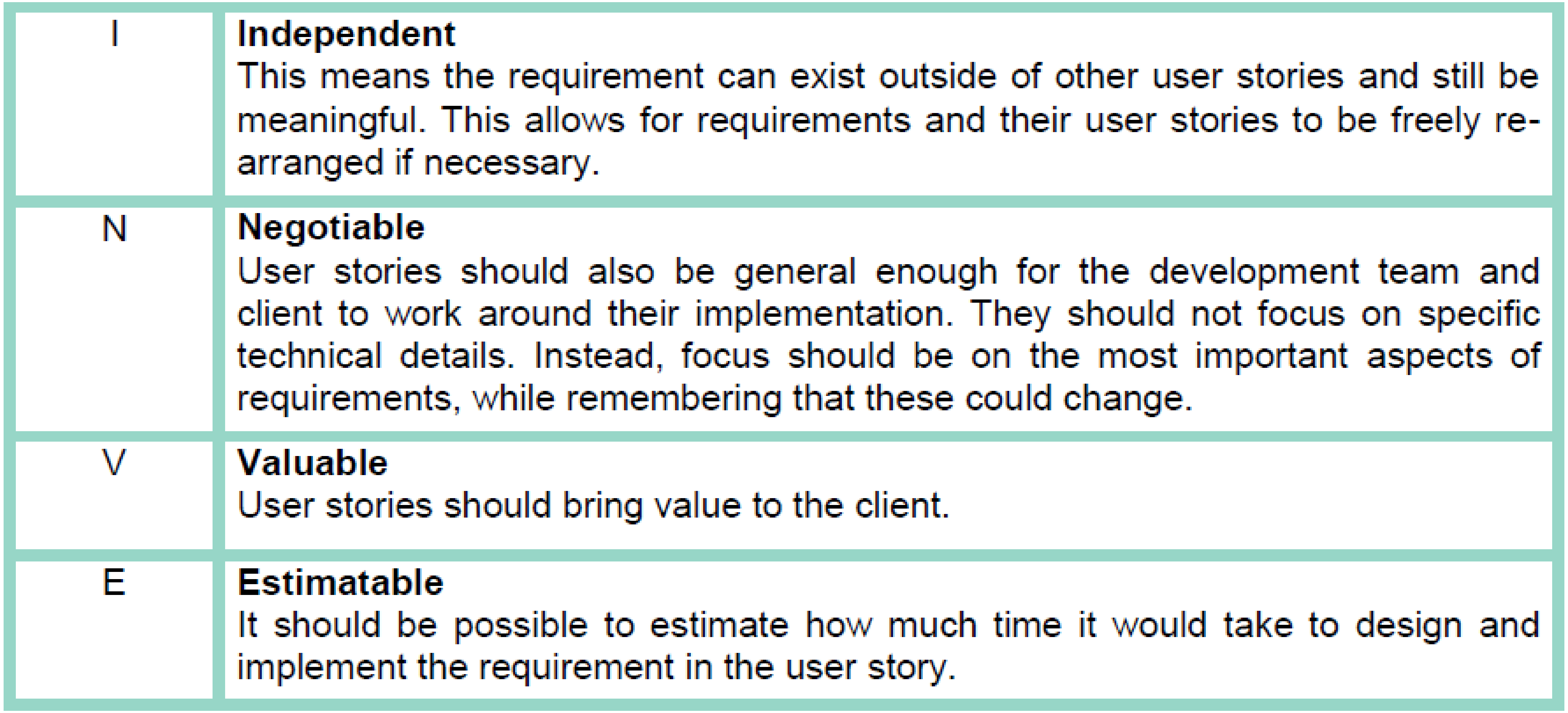
- Given a logged-in user viewing a gallery
- when an image the user has rated is displayed
- then the rating widget should reflect the user's previously entered rating

Story Example: Third Acceptance Criteria



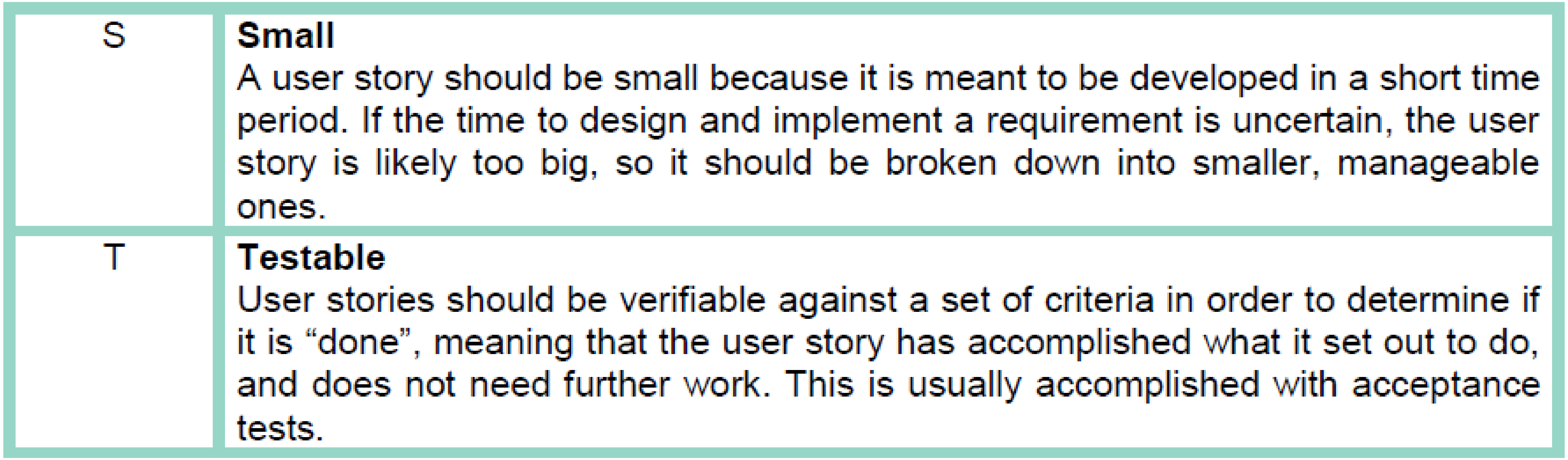
- Given a logged-in user viewing a gallery
- when the "Favorites" switch is toggled on
- then only images the user has rated should be shown, in descending order of rating

A Good User Story



I	**Independent** This means the requirement can exist outside of other user stories and still be meaningful. This allows for requirements and their user stories to be freely re-arranged if necessary.
N	**Negotiable** User stories should also be general enough for the development team and client to work around their implementation. They should not focus on specific technical details. Instead, focus should be on the most important aspects of requirements, while remembering that these could change.
V	**Valuable** User stories should bring value to the client.
E	**Estimatable** It should be possible to estimate how much time it would take to design and implement the requirement in the user story.

A Good User Story …



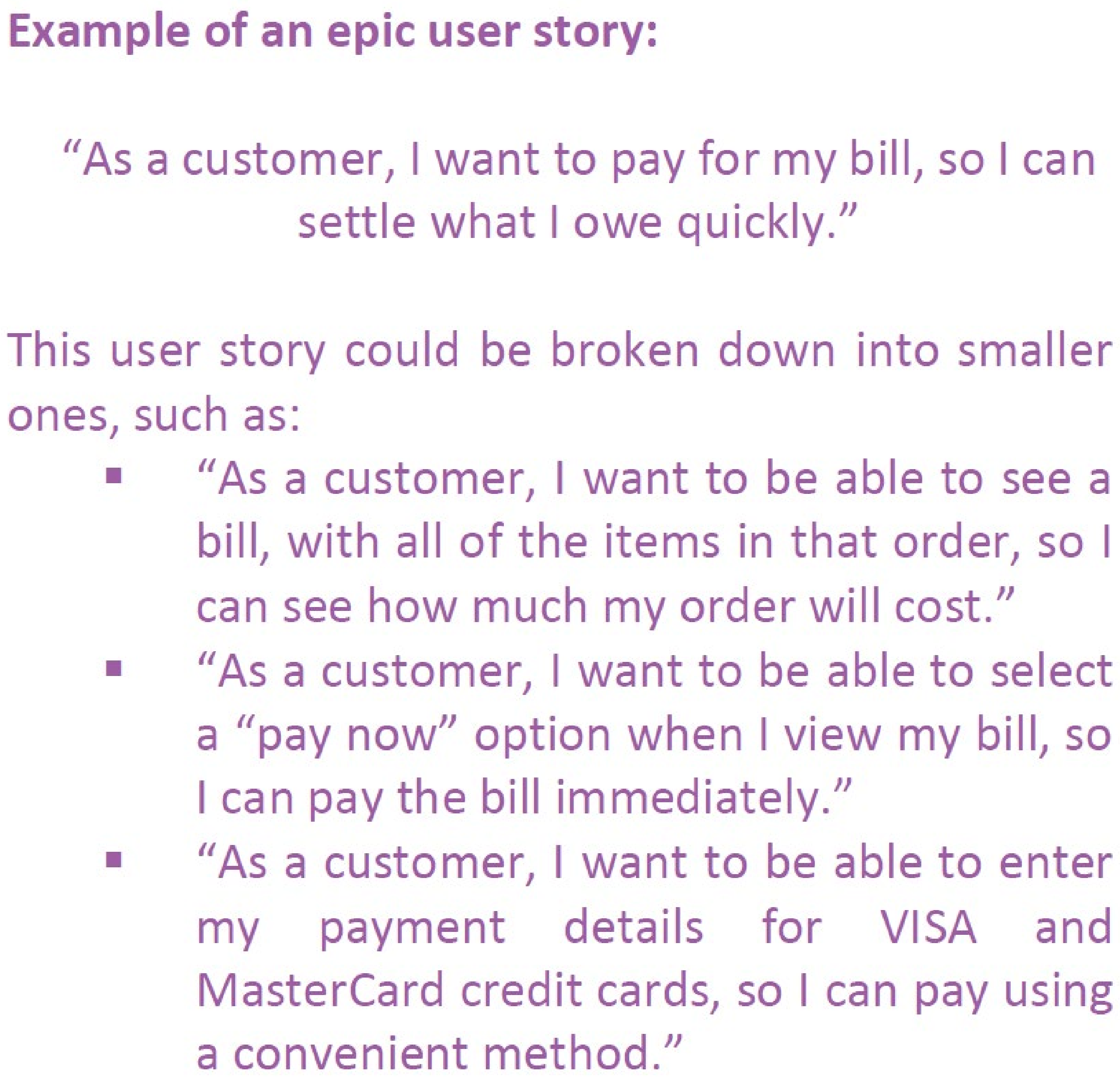
S	**Small** A user story should be small because it is meant to be developed in a short time period. If the time to design and implement a requirement is uncertain, the user story is likely too big, so it should be broken down into smaller, manageable ones.
T	**Testable** User stories should be verifiable against a set of criteria in order to determine if it is “done”, meaning that the user story has accomplished what it set out to do, and does not need further work. This is usually accomplished with acceptance tests.

What You Will Learn

- ✓ Scrum Rituals: Daily Standups & Sprint Planning
- ✓ Scrum Artifacts: User Stories

Appendix: Breaking Down an Epic User Story

An Epic User Story should break down into smaller ones



Example of an epic user story:

"As a customer, I want to pay for my bill, so I can settle what I owe quickly."

This user story could be broken down into smaller ones, such as:

- "As a customer, I want to be able to see a bill, with all of the items in that order, so I can see how much my order will cost."
- "As a customer, I want to be able to select a "pay now" option when I view my bill, so I can pay the bill immediately."
- "As a customer, I want to be able to enter my payment details for VISA and MasterCard credit cards, so I can pay using a convenient method."

Appendix: Research phase

Whether you are creating a landing page, a complex website, or an entire ecosystem for a product, the research phase is very important for structuring a functional design solution.

After analyzing the brief and understanding your stakeholder's vision, you need to set up a mission statement.

Mission statement is a short and affirmative description of the project’s purpose that highlights the problem your product is going to solve.

Appendix: Mission Statement Example

Setting up a mission statement helps you and your team always keep a product's purpose in mind, reducing the risk of going off the rails.

Mission Statement for Streaming Service

An easy-to-use, modern streaming service that allows users to browse a collection of culturally interesting content, such as documentaries and movies for entertainment and educational purposes, for free.

Access to Course Resources

https://github.com/samanipour/Practical-Agile-for-Software-Product-Management.git

References

[1] Green, M. D. (2016). Scrum: Novice to Ninja: Methods for Agile, Powerful Development, SitePoint.

[2] Ockerman, S. and S. Reindl (2019). Mastering professional scrum: A practitioner’s guide to overcoming challenges and maximizing the benefits of agility, Addison-Wesley Professional.

[3] Martin, R. C. (2019). Clean Agile, Pearson Education.

[4] Hall, G. M. (2017). Adaptive Code: Agile coding with design patterns and SOLID principles, Microsoft

[۵] سامانی‌پور، علی. (۲۰۱۸). آموزش اسکرام برای مدیریت چابک فرایند توسعه اپلیکیشن های وب و موبایل، فرادرس