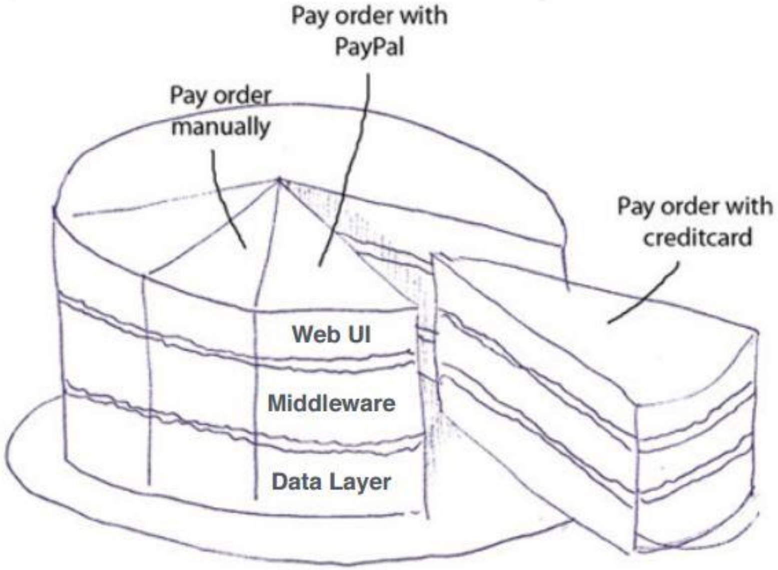
User Story Splitting Patterns
Vertical Slices

over

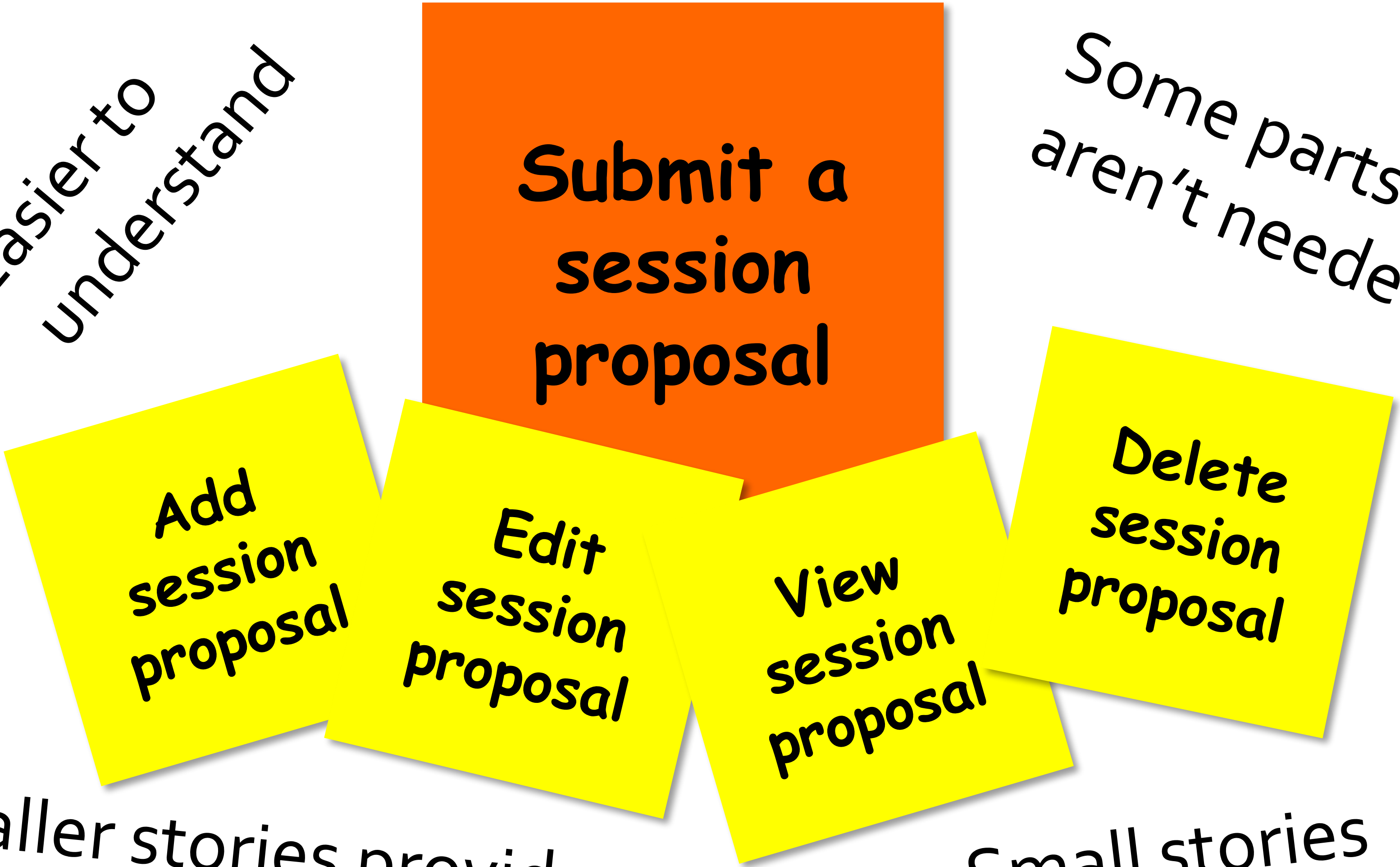
Horizontal Slices



Why Split User Stories?

Easier to understand

Some parts aren't needed



Submit a session proposal

Add session proposal

Edit session proposal

View session proposal

Delete session proposal

Smaller stories provide faster feedback

Small stories increase progress

Split Conditions

What This Is

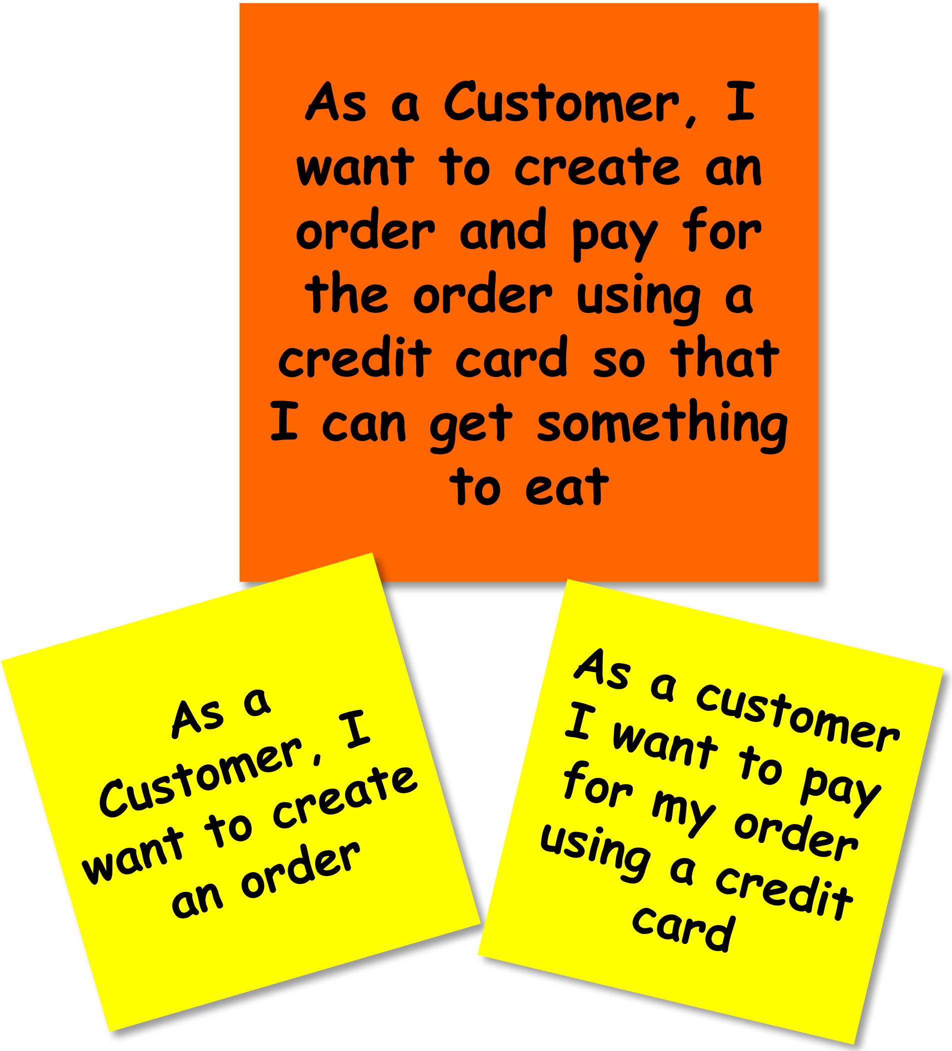
If a story has multiple items listed in “what” make each a separate story

When to Use it

Words such as “and” or “or” appear

Helpful questions

Are all of these conditions necessary (right now)?



As a Customer, I want to create an order and pay for the order using a credit card so that I can get something to eat

As a Customer, I want to create an order

As a customer I want to pay for my order using a credit card

Workflow

What This Is

Identify the specific steps in a workflow. Implement the workflow in stages, creating multiple stories for the incremental implementation

When to Use it

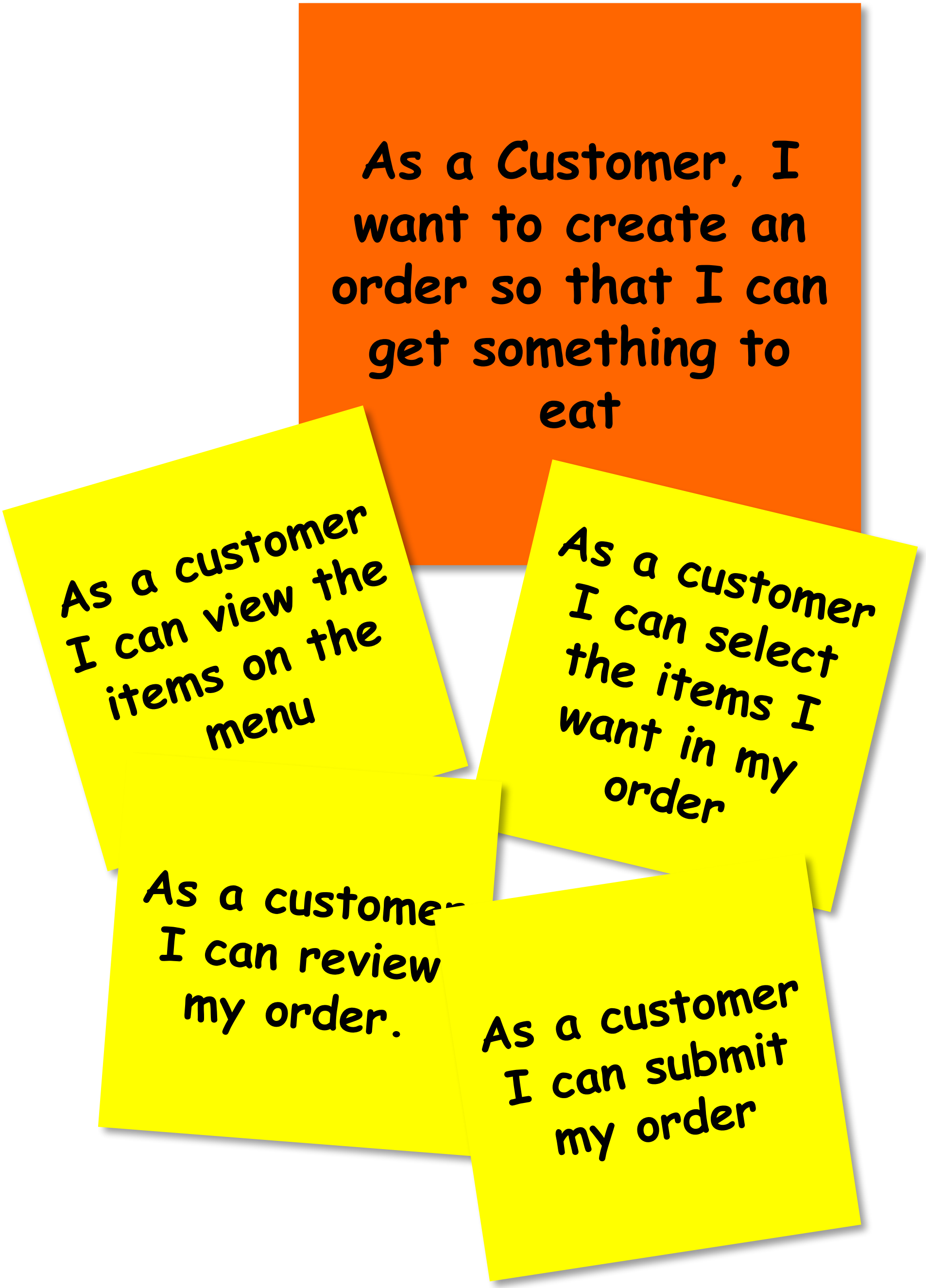
The initial story describes a workflow or process.

Helpful questions

What steps does a user perform?

Are all steps necessary (right now)?

Can steps be simplified (for now)?



As a Customer, I want to create an order so that I can get something to eat

As a customer I can view the items on the menu

As a customer I can select the items I want in my order

As a custome I can review my order.

As a customer I can submit my order

Use Case Scenarios

What This Is

Similar to the Workflow pattern, one story represents the happy path, different stories for alternate paths.

When to Use it

The initial story refers to an interaction

described by a use case

Helpful questions

What does the happy/alternate flow look like?

Are all the alternate flows necessary (right now)?

Can the alternate flows be simplified (right now)?

As a Customer, I want to create an order so that I can get something to eat

As a customer I want to order one item.

As a customer I want to order multiple items.

As a customer I want to cancel my order.

Operations

What This Is

Splitting a story based on the different operations done on an entity (Create, Read, Update, Delete).

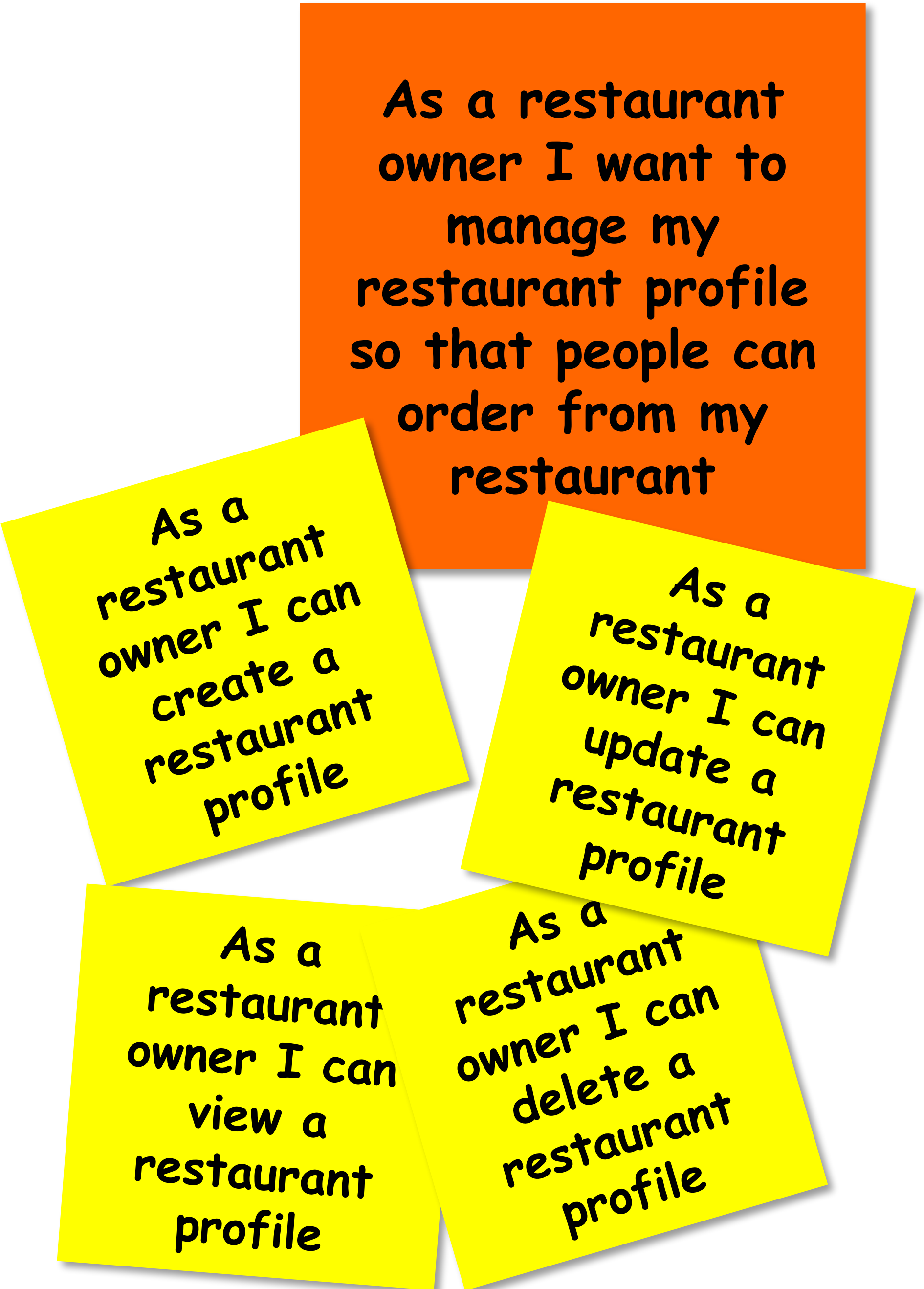
When to Use it

The story is about managing or configuring something

Helpful questions

What operations does the story entail?

Are all the operations necessary (right now)?



As a restaurant owner I want to manage my restaurant profile so that people can order from my restaurant

As a restaurant owner I can create a restaurant profile

As a restaurant owner I can update a restaurant profile

As a restaurant owner I can view a restaurant profile

As a restaurant owner I can delete a restaurant profile

Zero > One > Many

What This Is

Splitting a story that deals with multiple instances of an entity into stories that deal with one instance, and then multiple instances.

When to Use it

You are working with an entity where multiple instances are allowed.

Helpful questions

Do we need to be able to handle multiple instances, or is one instance sufficient?

As a restaurant owner, I want to view a customer's order.

As a restaurant owner, I want to view an order with no menu items.

As a restaurant owner I want to view an order with 1 menu item.

As a restaurant owner I want to view an order with multiple items

Core & Enhance

What This Is

Splitting a story to deal with the simple core conditions and then enhancing the functionality with additional user stories.

When to Use it

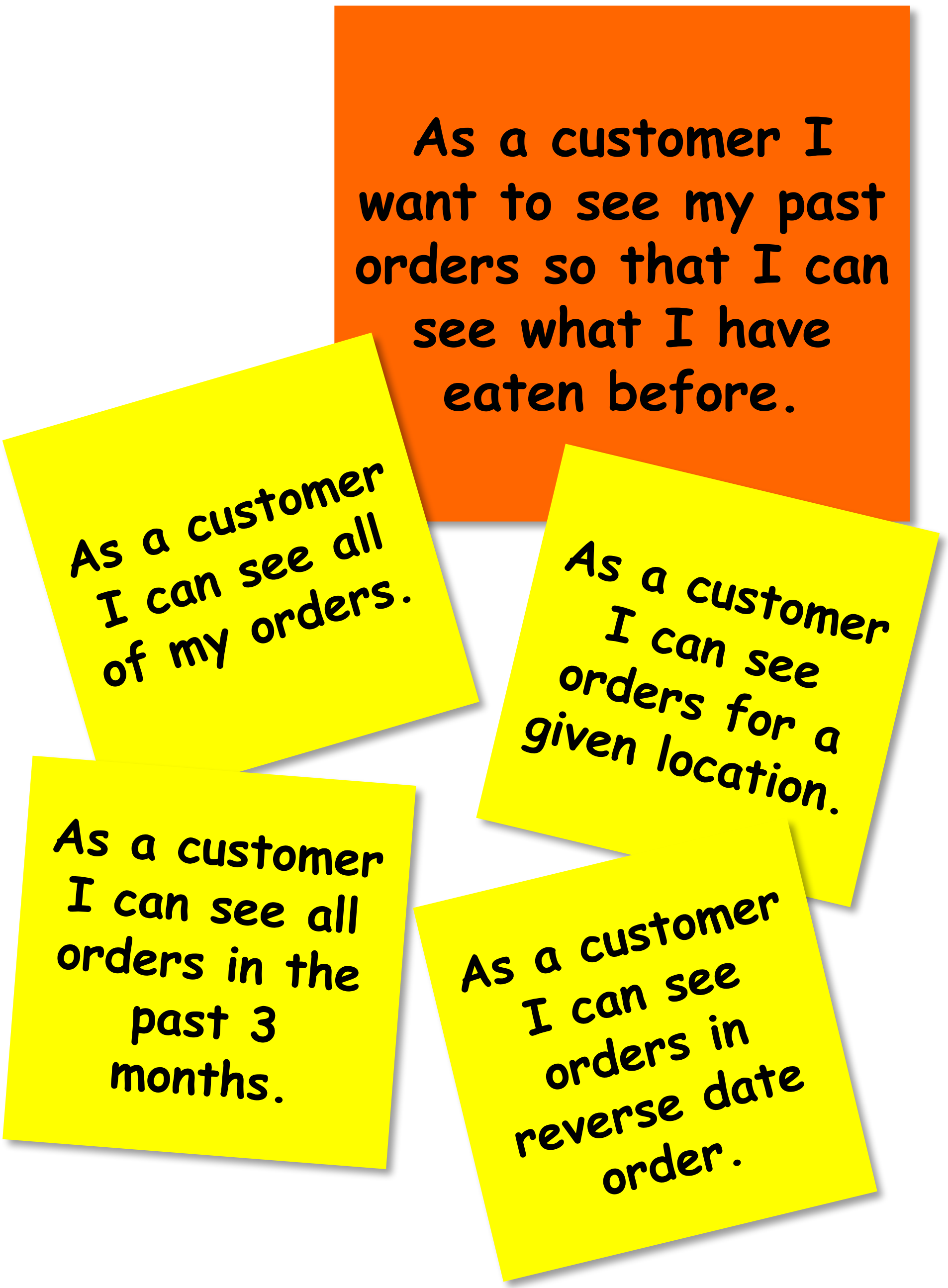
When the story has a simple core that provides most of the learning.

Helpful questions

What's the simplest version of this?

What data types are supported?

What parameters are relevant?



As a customer I want to see my past orders so that I can see what I have eaten before.

As a customer I can see all of my orders.

As a customer I can see orders for a given location.

As a customer I can see all orders in the past 3 months.

As a customer I can see orders in reverse date order.

Major Effort

What This Is

Splitting a story in a way that requires substantial effort for the first story and less effort for all subsequent stories.

When to Use it

When you apply the most obvious split, any story you do first is the most difficult.

Helpful questions

Can we create the necessary infrastructure and deliver value at the same time?

Does it make a difference which option we do first?

As a customer I can pay for my order with Amex, MC, Visa, or DC so that I don’t have to deal with cash..

As a customer I can pay with one credit card type (Amex, MC, Visa, DC)

As a customer I can pay with all credit card types, given one card type already done.

Acceptance Criteria

What This Is

Split a user story into multiple stories to handle different scenarios in the Acceptance Criteria.

When to Use it

When the team identifies multiple scenarios for a story, or accurate, but not entirely relevant acceptance criteria identified.

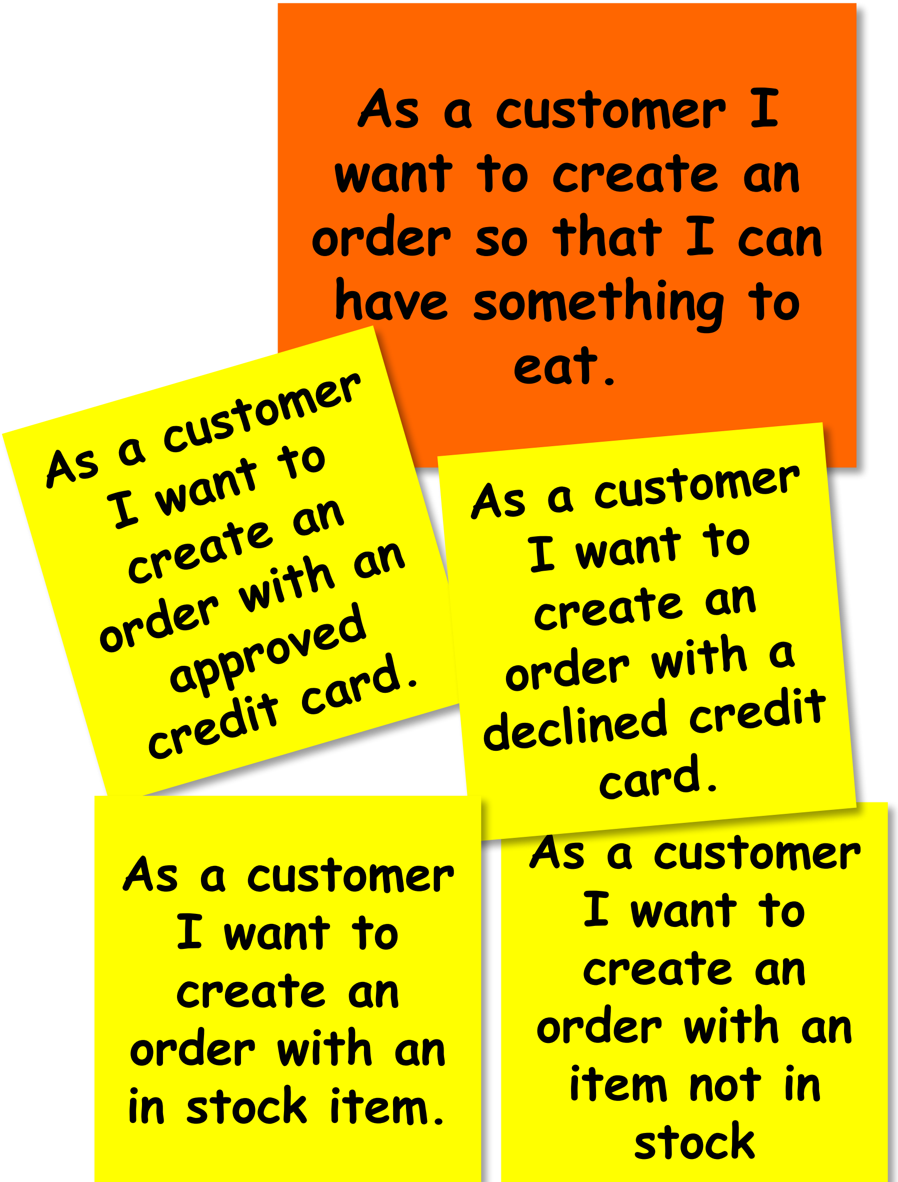
Helpful questions

What tests are used to verify this story?

What acceptance criteria apply?

What scenarios are relevant?

Are all test scenarios necessary (right now?)



As a customer I want to create an order so that I can have something to eat.

As a customer I want to create an order with an approved credit card.

As a customer I want to create an order with a declined credit card.

As a customer I want to create an order with an in stock item.

As a customer I want to create an order with an item not in stock

Variations in Data

What This Is

The story does the same thing to different types of data. Create a story for each option.

When to Use it

When a solution has to support multiple options.

Helpful questions

Are these options necessary right now?

What is the most common option we need to care for right away?

As a customer I want to view the menu in my native language so I can decide what I want to eat.

As a customer I want to view the menu in English.

As a customer I want to view the menu in French.

As a customer I want to view the menu in Spanish.

As a customer I want to view the menu in Swahili

Data Boundaries

What This Is

Splitting a story dealing with several attributes of the same entity.

Support a few key elements first and introduce remaining data elements later via additional stories.

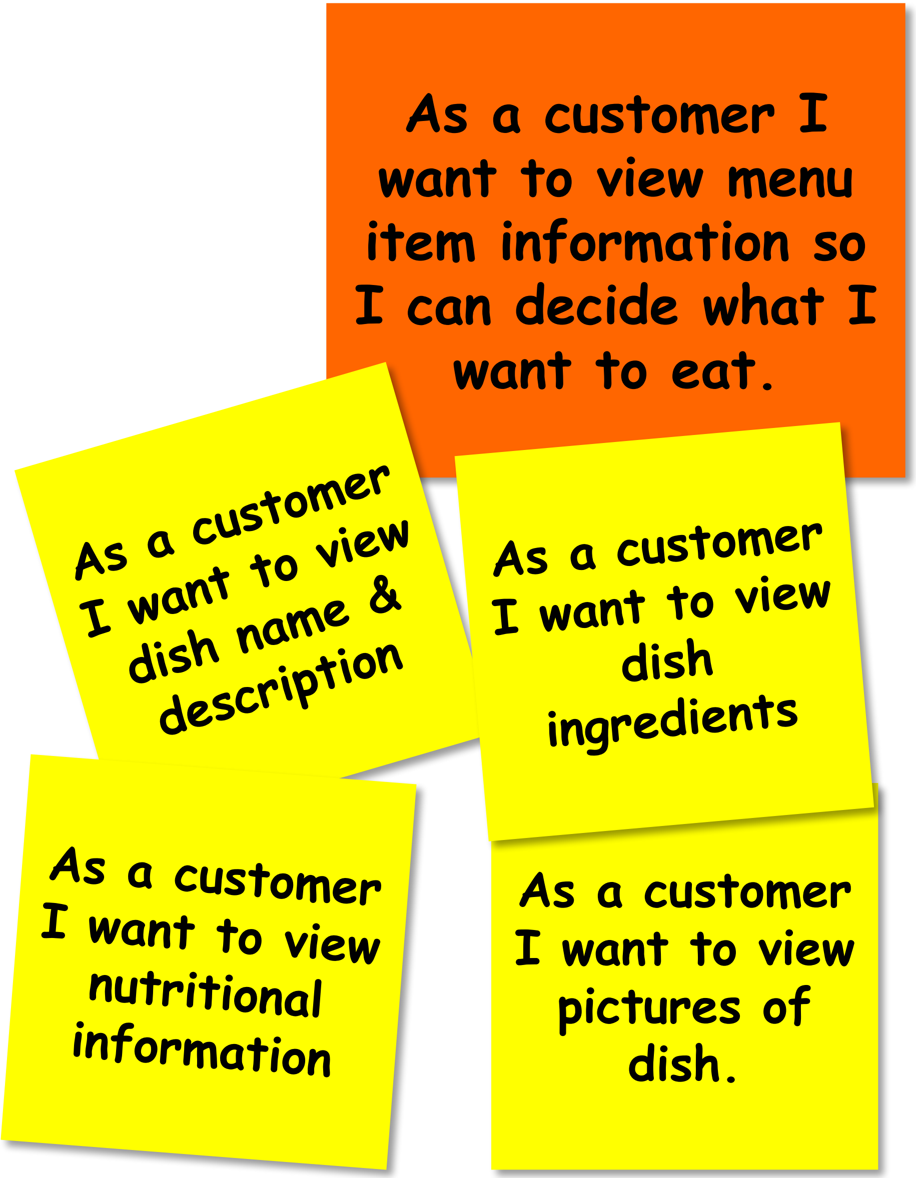
When to Use it

When a story is dealing with several different pieces of data.

Helpful questions

What are the essential data elements we need to have?

What data elements are not necessary right now?



As a customer I want to view menu item information so I can decide what I want to eat.

As a customer I want to view dish name & description

As a customer I want to view dish ingredients

As a customer I want to view nutritional information

As a customer I want to view pictures of dish.

Interface Variations

What This Is

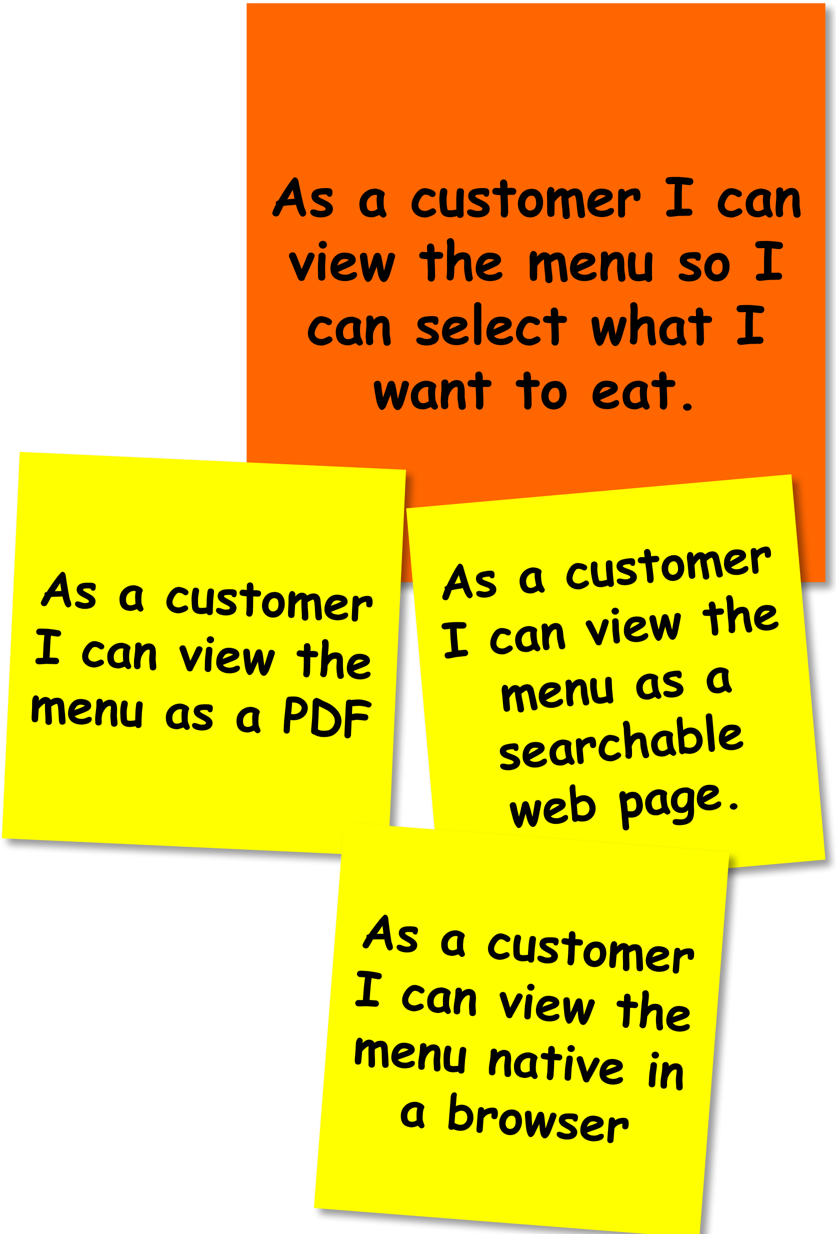
Splitting a story dealing with a complex interface with addition stories to incrementally add in complexity.

When to Use it

When the user story deals with a complex user interface where a simpler one will work in the meantime

Helpful questions

What is the simplest user interface we can use?



As a customer I can view the menu so I can select what I want to eat.

As a customer I can view the menu as a PDF

As a customer I can view the menu as a searchable web page.

As a customer I can view the menu native in a browser

Platform Options

What This Is

Split a story adding a new user interface by the various platforms that are applicable.

When to Use it

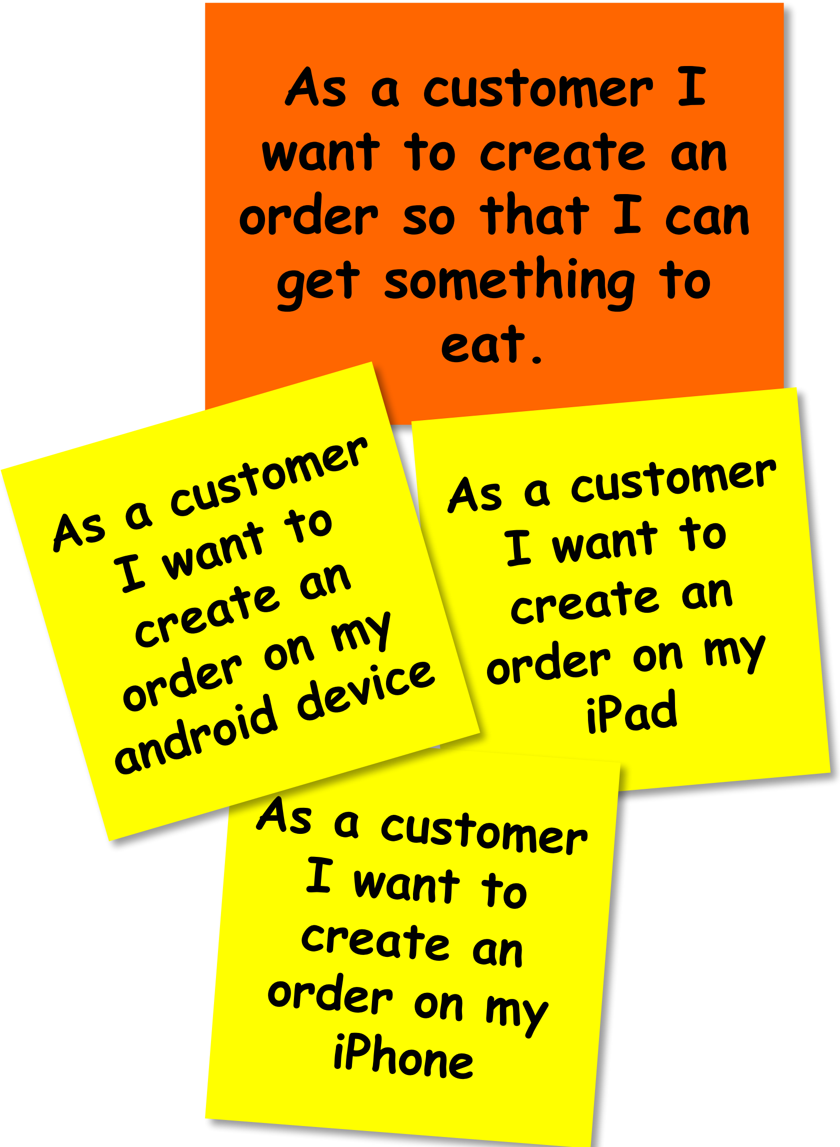
When adding a new user interface that may be accessed by multiple different platforms.

Helpful questions

Which platforms are supported?

Are all platforms required (right now)?

Are some platforms more difficult to support than others?



As a customer I want to create an order so that I can get something to eat.

As a customer I want to create an order on my android device

As a customer I want to create an order on my iPad

As a customer I want to create an order on my iPhone

Business Rules

What This Is

Split a user story so that complex business rules are handled by separate business rules.

A special case of this is deferring input validation to a later story.

When to Use it

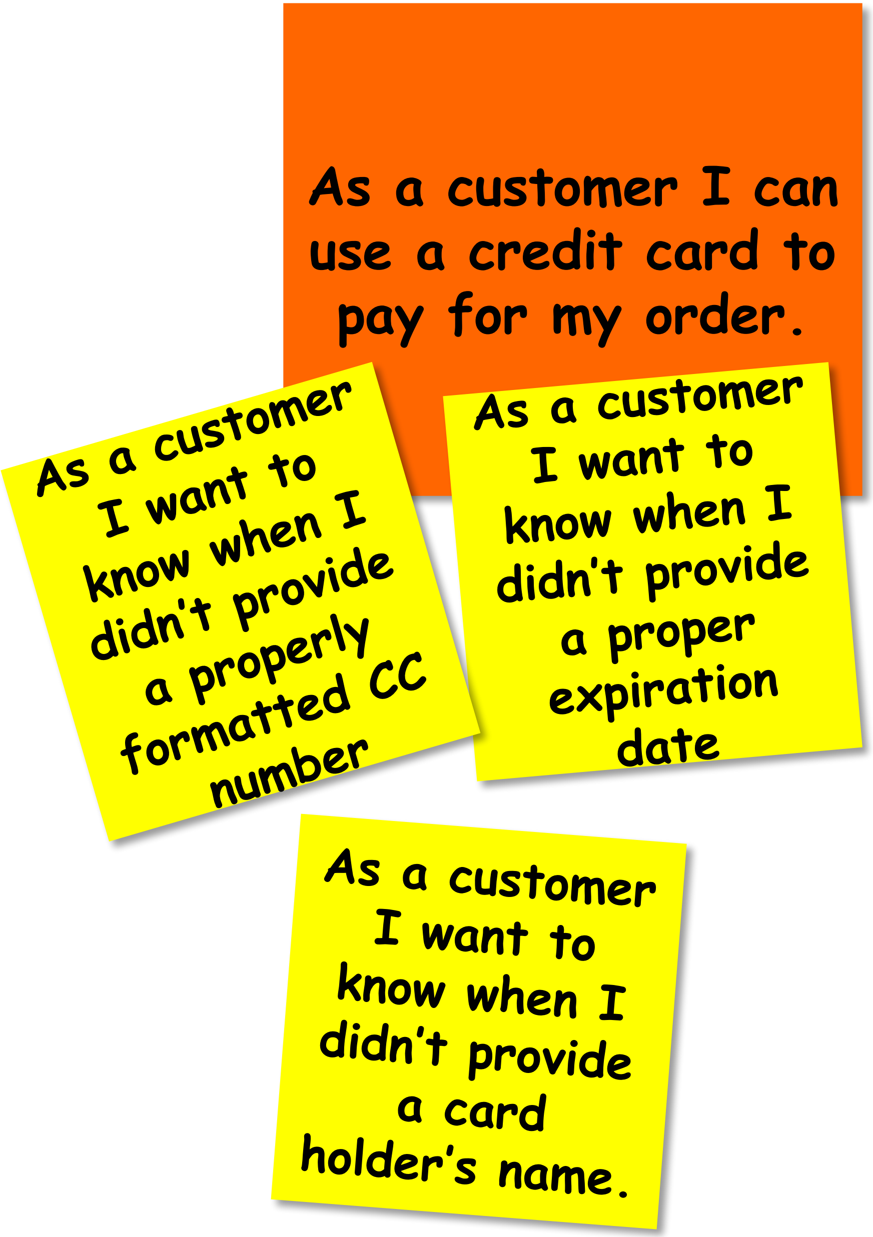
When a story has a variety of business rules, often identified by a large number of examples.

Helpful questions

What rules apply to this story?

Are all of those rules necessary (right now)?

Can simpler rules suffice (for now)?



As a customer I can use a credit card to pay for my order.

As a customer I want to know when I didn't provide a properly formatted CC number

As a customer I want to know when I didn't provide a proper expiration date

As a customer I want to know when I didn't provide a card holder's name.

Role

What This Is

Split a story so that the functionality is implemented at different times for different roles.

When to Use it

When a story impacts multiple roles and the impact is different for each role.

Helpful questions

What roles are involved in this story?

Are any roles necessary now?

As a customer I want to create an order so I can get something to eat.

As a customer I want to create an order.

As a restaurant owner, I want to create an order for a customer

Defer System Qualities

What This Is

Split the story to deliver the necessary functionality first, then add additional stories to improve performance, scalability, usability or precision

When to Use it

When the base functionality does not exist at all, initial implementation is not that difficult, the team can learn a lot from it, and the hard work is in making it better.

Helpful questions

Is it necessary for this to have optimal performance (right now)?

As a customer I want to know that the restaurant received my order so that I'm sure I'll get some food.

As a customer I want to know that the restaurant received my order.

As a customer I want to know that the restaurant received my order within 5 seconds.

Spikes

What This Is

Split the story to allow for some research and investigation on functionality before implementing it.

When to Use it

When the team finds they are uncertain about the implementation of a story and they need to do some research.

Helpful questions

What are the 1 – 3 questions you have about the story?

As a customer I want to see the menu in the language of my choice so that I can pick something I want.

Spike: investigate translation functionality.

Low Fidelity/High Fidelity

What This Is

Split the story into a gradual increase of quality.

When to Use it

When getting to the optimal level of quality or usability is too expensive to deliver a solution immediately.

Helpful questions

What is good enough for this functionality?

As a customer I would like a recommendation of what food to order so that I can know I'm getting a good dish.

As a customer I want to see a numeric rank of top selling menu items.

As a customer I want to see the aggregate rankings of other customers

Transient then Persistent

What This Is

Split stories based on actions to pass data along and those necessary to save data.

When to Use it

When the story deals with functionality that includes storing data, but does not require storing data.

Helpful questions

Do we have to store this data (right now)?

As a customer I want to use a credit card to pay for my order so that I don't have to carry cash.

Collect credit card info and use it to gain authorization, but not store it.

As a customer, I want to save my credit card information to my profile.

Dummy then Dynamic Data

What This Is

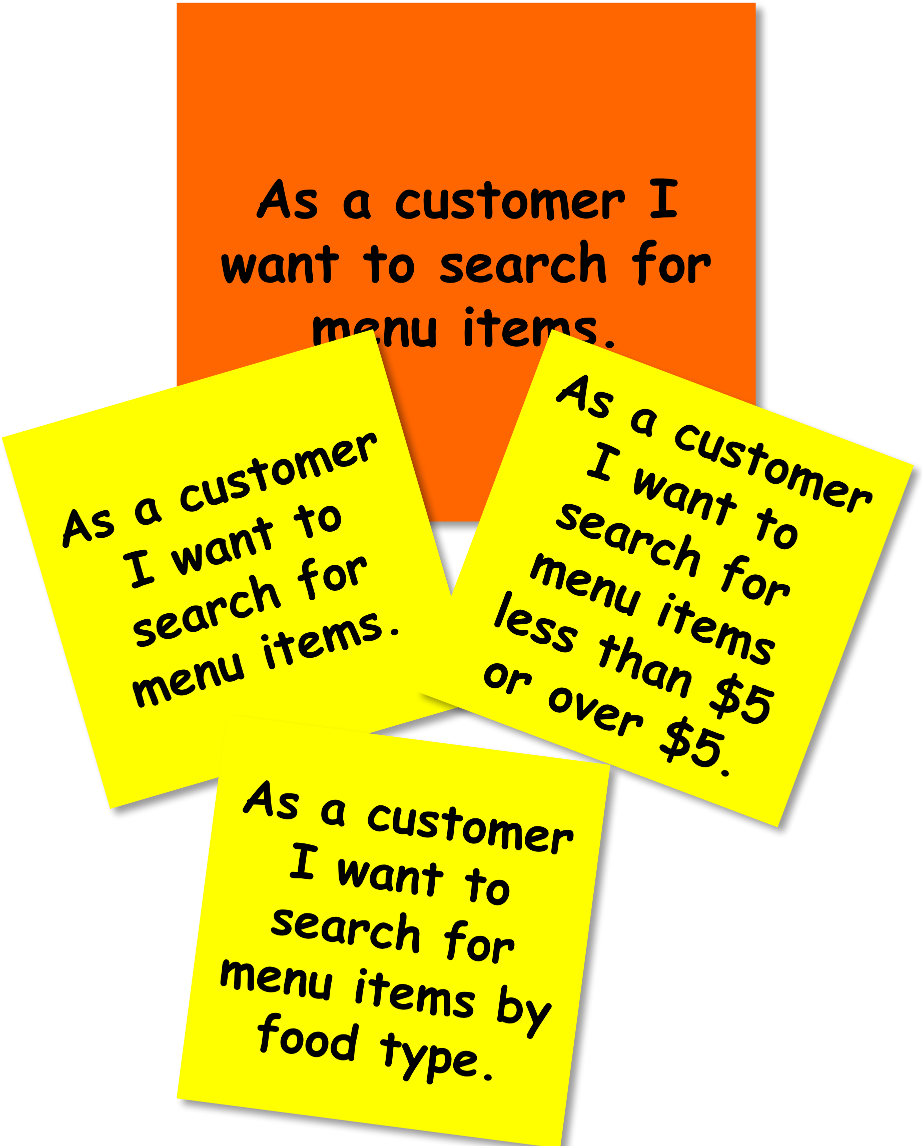
Split stories to first use static data and then add stories to make the data dynamic.

When to Use it

When the story covers parameters where the values differ depending on conditions.

Helpful questions

Does the data need to be dynamic (right now)?



As a customer I want to search for menu items.

As a customer I want to search for menu items.

As a customer I want to search for menu items less than $5 or over $5.

As a customer I want to search for menu items by food type.

Manual then Automated

What This Is

Split stories that utilize an existing manual process. Implement the functionality that does not exist.

When to Use it

When functionality you are developing interacts with an existing manual process, develop the functionality but in the short term, do not automate the manual process.

Helpful questions

Is it necessary to automate the manual process right now?

As a customer I want to use multiple credit cards to pay for my order.

As a customer I want to use a credit card to pay for my order.*

*Use manual process to deal with multiple cards

Defer Error Handling or Logging

What This Is

Split story so that you focus on functionality first, then go back and provide functionality to log errors and handle them.

When to Use it

When you need to gain feedback on functionality quickly to determine whether to keep it.

Helpful questions

Is it necessary to deal with exceptions for this functionality (right now)?

As a restaurant owner I want to authorize the customer's credit card payment..

As a restaurant owner I want to know when the authorization process encountered errors

As a restaurant owner I want to know what errors the authorization process encountered.

Which Pattern to Use

Since there are multiple patterns that can be used to split the same story, here are three rules of thumb to guide your selection:

1. Choose the split that **lets you deprioritize** or throw away a story.

2. Choose the split that gets you **more equally sized** small stories.

3. Chose the split that eliminates or at least **reduces dependencies**

http://www.agileforall.com/2009/10/patterns-for-splitting-user-stories/

Additional References

- Splitting User Stories Presentation http://www.slideshare.net/arsenalist/splitting-userstories
- Splitting User Stories Cheat sheet https://twitter.com/chrisverwijs/status/335109871802384385
- Breaking Down Larger Stories http://agileinaflash.blogspot.com/2009/02/breaking-down-larger-stories.html
- Ways to Split User Stories: http://lassekoskela.com/thoughts/7/ways-to-split-user-stories/
- User Story Hamburger technique http://gojko.net/2012/01/23/splitting-user-stories-the-hamburger-method/
- Features to User Stories http://idiacomputing.com/pub/UserStories.pdf
- Patterns for Splitting User Stories http://www.agileforall.com/2009/10/patterns-for-splitting-user-stories/
- Twenty Ways to Split Stories http://xp123.com/articles/twenty-ways-to-split-stories/