



# User Story

CSC 233 - Spring 2018

Behnam Salamati  
William Dobson



# What is a user story?

User stories are short, simple descriptions of a feature told from the perspective of the person who desires the new capability, usually a user or customer of the system.

## Stories are:

1. User's needs
2. Product descriptions
3. Planning items
4. Tokens for a conversation
5. Mechanisms for deferring conversation



# User Stories, Epics and Themes

- A user story is simply something a user wants.
- A Scrum epic is a large user story. There's no magic threshold at which we call a particular story an epic. It just means "big user story."
- "Theme" is a collection of user stories.



# Who writes user stories?

Anyone can write user stories. It's the product owner's responsibility to make sure a product backlog of agile user stories exists, but that doesn't mean that the product owner is the one who writes them.



# What are three aspects of user stories?

1. **Card** - A written description of the user story for planning purposes and as a reminder
2. **Conversation** - A section for capturing further information about the user story and details of any conversations
3. **Confirmation** - A section to convey what tests will be carried out to confirm the user story is complete and working as expected



# User Story Template – 1

- As a [user role] I want to [goal]  
so I can [reason]
- As a [type of user] I want to [perform some task] so that I can [reach some goal]

Example:

- As a registered user I want to log in  
so I can access subscriber-only content



## User Story Template – 2

- Who (user role)
- What (goal)
- Why (reason)
  1. gives clarity as to why a feature is useful
  2. can influence how a feature should function
  3. can give you ideas for other useful features
  4. that support the user's goals



## Why use User Stories?

- Comprehensible by both stakeholders and the development team
- Emphasize verbal rather than written communication
- Right size for planning, works for iterative development
- Defer detail until you have the best understanding you are going to have about what you really need
- Focus on user goals rather than feature attributes
- Stories support evolutionary development





# How detailed should a User Story be?

## Detailed enough

- For the team to start working from
- To establish further details and clarifications at the time of development.



# Techniques for creating a set of stories

- User interviews
- Questionnaires
- Observation
- Story-writing workshops



# What is story-writing workshop?

A story-writing workshop is a meeting that includes developers, users, the product customer and other parties who can contribute by writing stories. During the workshop the participants write as many stories as possible.



# Features of a good story

- Independent
- Negotiable
- Valuable to users or customers
- Estimatable
- Small
- Testable



# Independent

- Stories that depend on other stories are difficult to prioritize and estimate
- Example:
  - A company can pay for a job posting with a Visa card
  - A company can pay for a job posting with a Mastercard
  - A company can pay for a job posting with an American Express card



# Negotiable

- Story cards serve as reminders not contracts
- Details need to be fleshed out in conversation
- Story cards should have a phrase or sentence to serve as reminder to have conversation & notes about conversation



# Valuable

- Both to people using the software and paying for the software
- Avoid stories valued only by developers (make the benefits to customers/users apparent for these stories)
- Example
  - “All connections to the database are through a connection pool” could be rewritten as “Up to 50 users should be able to use the application with a 5-user database license”



# Estimable

- 3 common reasons why story might not be estimable
  - Developers lack domain knowledge
    - Get details from customer
  - Developers lack technical knowledge
    - Perform spike to explore technology
  - Story is too big
    - Split the story into smaller ones





# Small

- Easy to use in planning
- Split compound & complex stories
- Combine too small stories



# Testable

- Can't tell if story is done without tests
- Aim for most tests to be automatable



# Compound and Complex User Story

- A compound story is an epic that comprises multiple shorter stories.
- Unlike the compound story, the complex story is a user story that is inherently large and cannot easily be disaggregated into a set of constituent stories.



# Splitting Stories

- Compound
  - Conversations may reveal multiple stories
  - Split along Create/Update/Delete
  - Split along data boundaries
- Complex
  - Split into investigative and develop the new feature stories (define timebox for investigation)
- Make sure each split-off portion is a good story (INVEST)



# Splitting Large User Stories

1. The first approach is called **horizontal splitting** and involves breaking up user stories by the kind of work that needs to be done or the application layers that are involved. So, work that has to be done for the UI, databases, a front-end, a back-end and testing is split up into groups of tasks.
2. In Scrum (and I would say in general), **vertical splitting** is more useful. If a user story is split vertically, it is split in such a manner that the smaller items still generate some business value. The functionality will not be split across technical layers or tasks, but across functional layers.

# Example

## Automated Teller Machine (ATM) Horizontal and Vertical User Stories - Slicing the Cake

### Vertical User Stories

Cash Withdrawal (90% Usage)

Bank Statement

### Horizontal Stories

UI - PIN and Card Reader

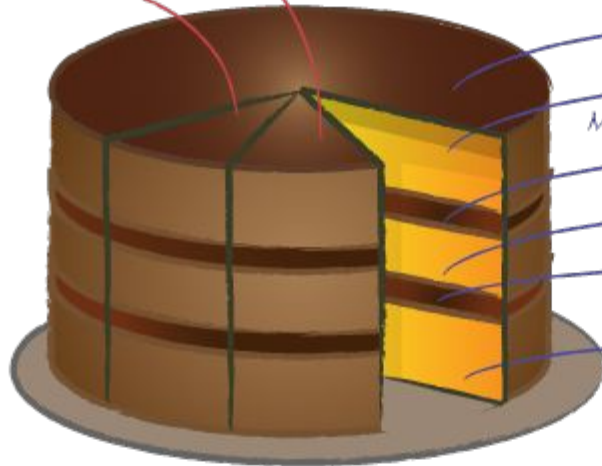
Security Layer

Middleware - Transaction Protocol

Tuxedo DB Interface

Transport Protocol

Bank Mainframe Database

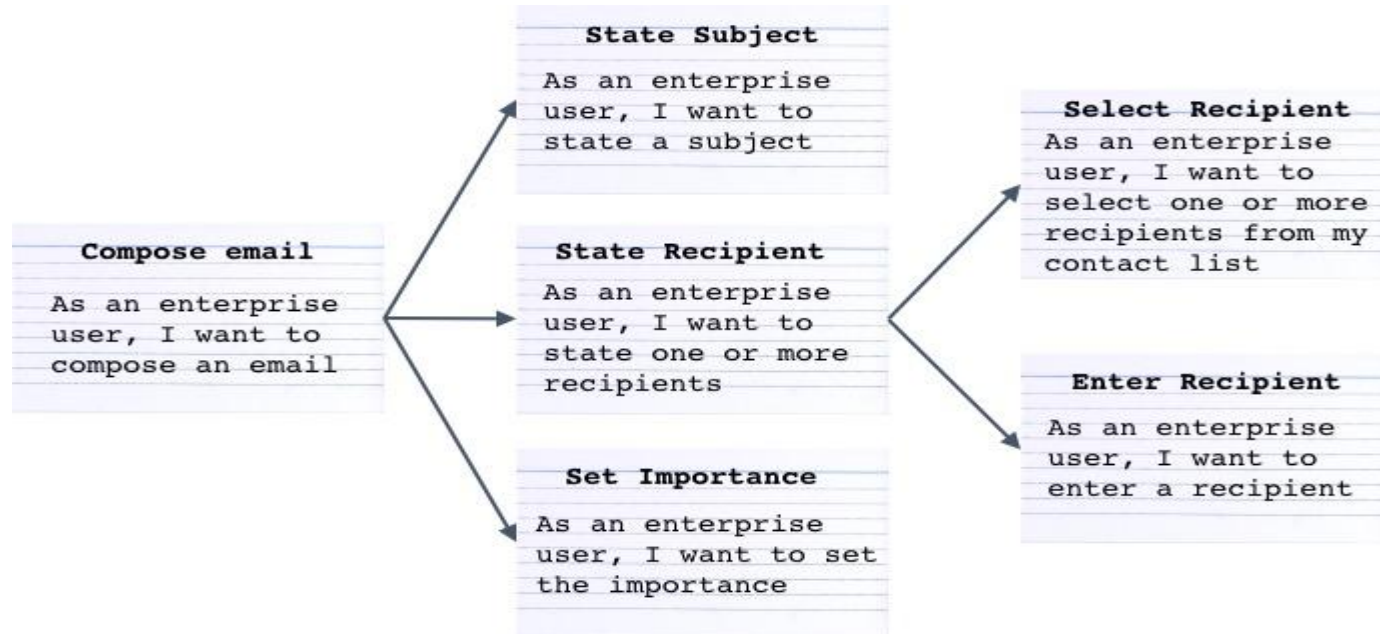




# Refining User Stories

Grooming the product backlog entails breaking down user stories from larger product backlog items until they are small enough to fit into a sprint. Although detailing product backlog items should be delayed until the last responsible moment, you might have to start refining a story a couple of sprints in advance before it can be implemented, particularly if the story is large or complex.

# Example







# What Are Story Points?

Story points are a unit of measure for expressing an estimate of the overall effort that will be required to fully implement a product backlog item or any other piece of work.



# Prioritizing User Stories with the MoSCoW Method

MoSCoW is a prioritisation technique for helping to understand and manage priorities. The letters stand for:

- Must Have
- Should Have
- Could Have
- Won't Have this time



# What Goes Into a Story Point?

Because story points represent the effort to develop a story, a team's estimate must include everything that can affect the effort. That could include:

- The amount of work to do
- The complexity of the work
- Any risk or uncertainty in doing the work

When estimating with story points, be sure to consider each of these factors.



# The definition of done

The definition of done (often called a “DoD”) establishes what must be true of each product backlog item for that item to be done. A typical DoD would be something similar to:

- The code is well written. (That is, we’re happy with it and don’t feel like it immediately needs to be rewritten.)
- The code is checked in. (Kind of an “of course” statement, but still worth calling out.)
- The code was either pair programmed or peer reviewed.
- The code comes with tests at all appropriate levels. (That is, unit, service and user interface.)
- The feature the code implements has been documented in any end-user documentation such as manuals or help systems.



# Consider Everything in the Definition of Done

A story point estimate must include everything involved in getting a product backlog item all the way to done. If a team's definition of done includes creating automated tests to validate the story (and that would be a good idea), the effort to create those tests should be included in the story point estimate.



# From user story to test case

We can also use templates to write test cases for the user stories. The template is as follows:

**Scenario:** a short description of the test scenario

**Given:** test preconditions

**When:** test action – input

**Then:** test result – output

**And:** can be used to include more than one precondition, input or output.



# Non-functional Requirements as User Stories

A common challenge with writing user stories is how to handle a product's non-functional requirements. These are requirements that are not about specific functionality ("As a user of a word processor, I want to insert a table into my document."), but are rather about an attribute or characteristic of the system.

Non-functional requirements can be considered as "constraints" we put on the system which fortunately constraints/non-functional requirements can be easily handled as user stories.



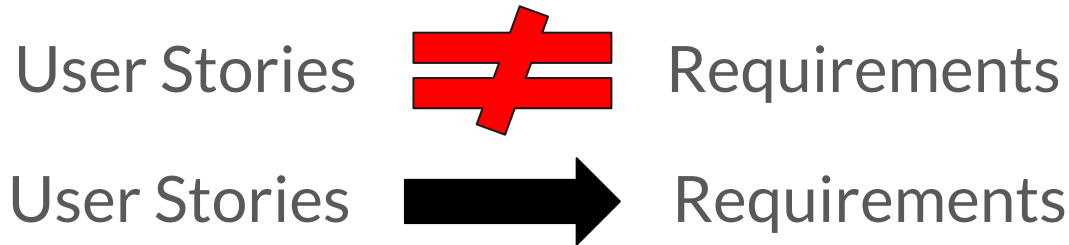
# Example

- As a customer, I want to be able to run your product on all versions of Windows from Windows 95 on.
- As the CTO, I want the system to use our existing orders database rather than create a new one, so that we don't have one more database to maintain.
- As a user, I want the site to be available 99.999 percent of the time I try to access it, so that I don't get frustrated and find another site to use.
- As someone who speaks a Latin-based language, I might want to run your software someday.

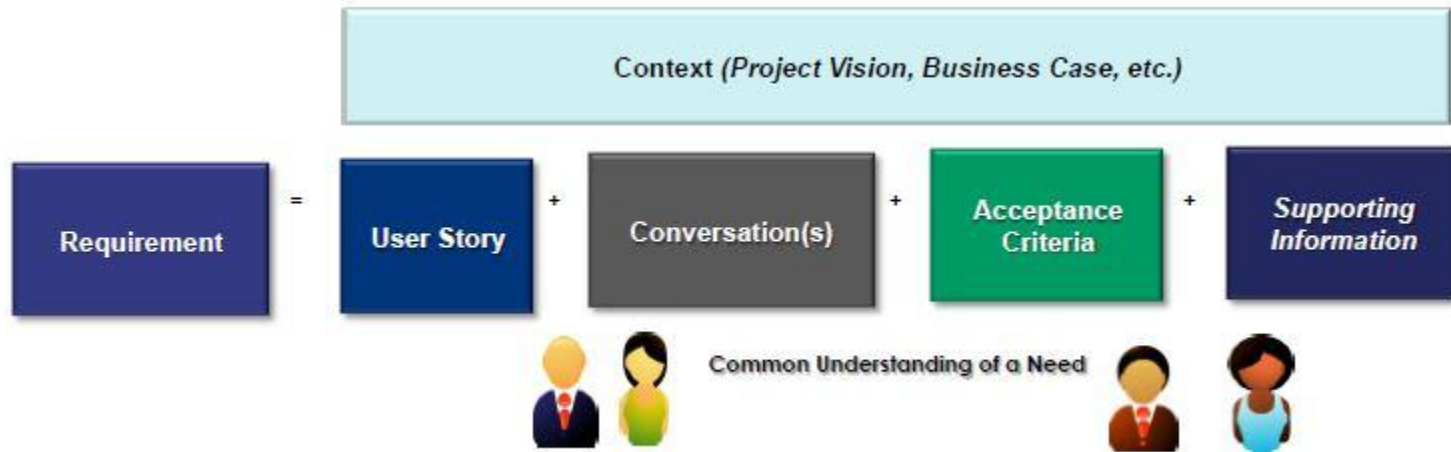




## What user stories are NOT



# Requirements, More than Just a Story

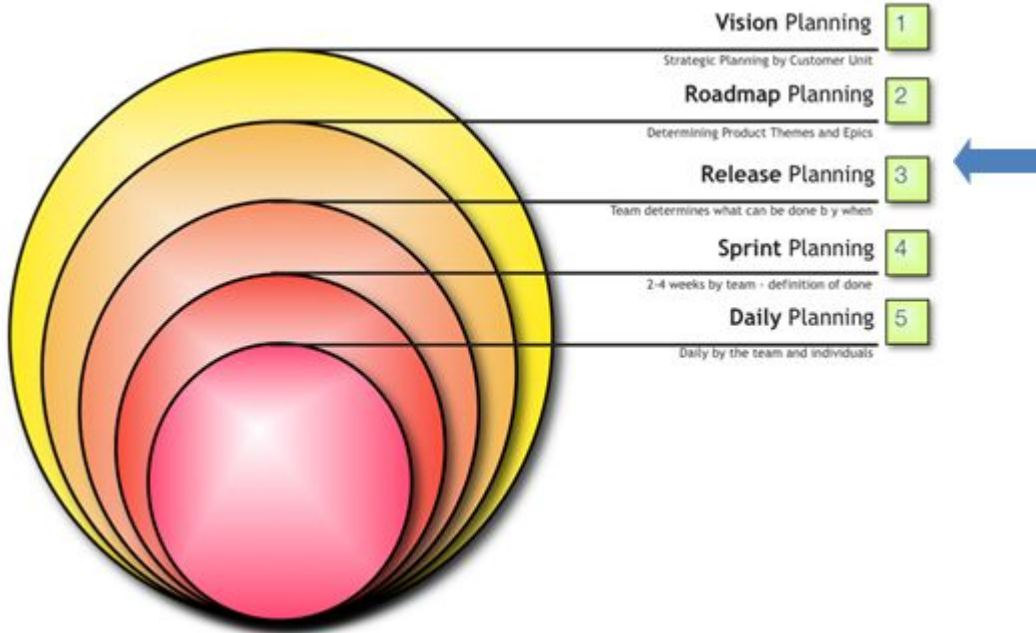




# Common pitfalls with writing user stories

- Forgetting about the User
- Too much Detail
- Lack of Conversation
- No Acceptance Criteria

# Where User Stories Fit In



**Must** have User Stories written, estimated & prioritized **prior to** Release Planning.



# Product Backlog

- A prioritized list of all user stories that may be delivered
- New items can be added at any time to the Product Backlog
- Items are defined and prioritized by Product Owner with input from others
- Team members estimate items in Product Backlog relative to each other using predetermined scale (story points)



## Non-User Stories

- Technology foundation stories
  - At times these can be stated in customer terms
- Dependencies from external teams
- Creative elements
- Spikes

As a developer, I want to upgrade to the latest version of the database software so that we have a supported product

Spike: As a developer, I need to investigate a semantic search algorithm to facilitate natural language searching of the person's financial record.



# User story map

User Story Mapping is an approach that is widely used by software companies for release planning. Story Map consists of user stories that are ordered in two independent dimensions. Horizontal axis represents the order of activities from the user's point of view, and they describe the behavior of the system. Down the vertical axis, it represents increasing sophistication of implementation.



# User story map

## Spatial arrangement:

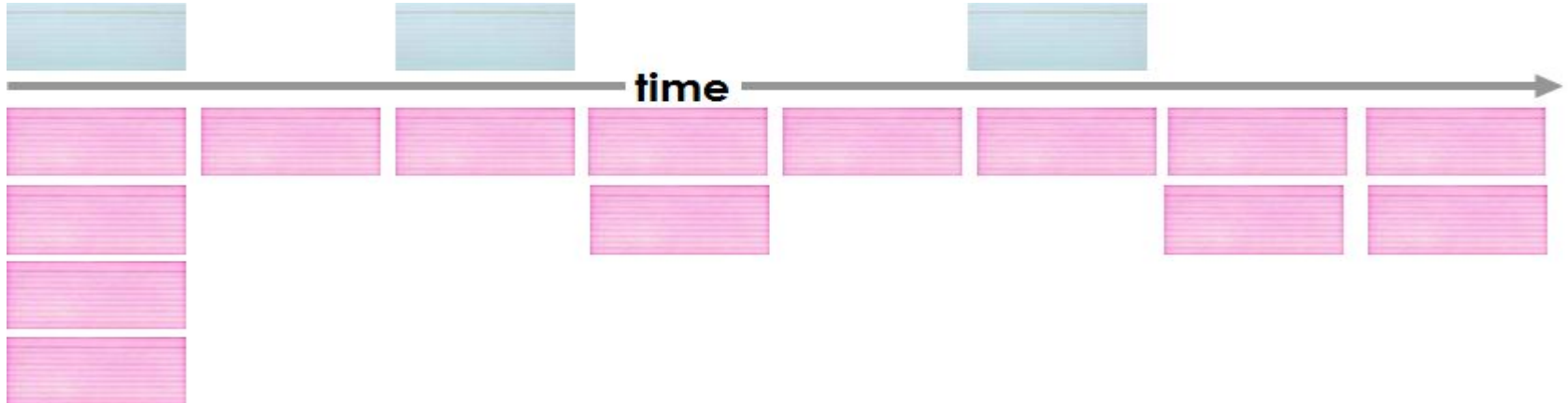
- By arranging activity and task-centric story cards spatially, we can identify bigger stories
- Arrange activities left to right in the order you'd explain them to someone when asked the question: "What do people do with this system?"





# User story map

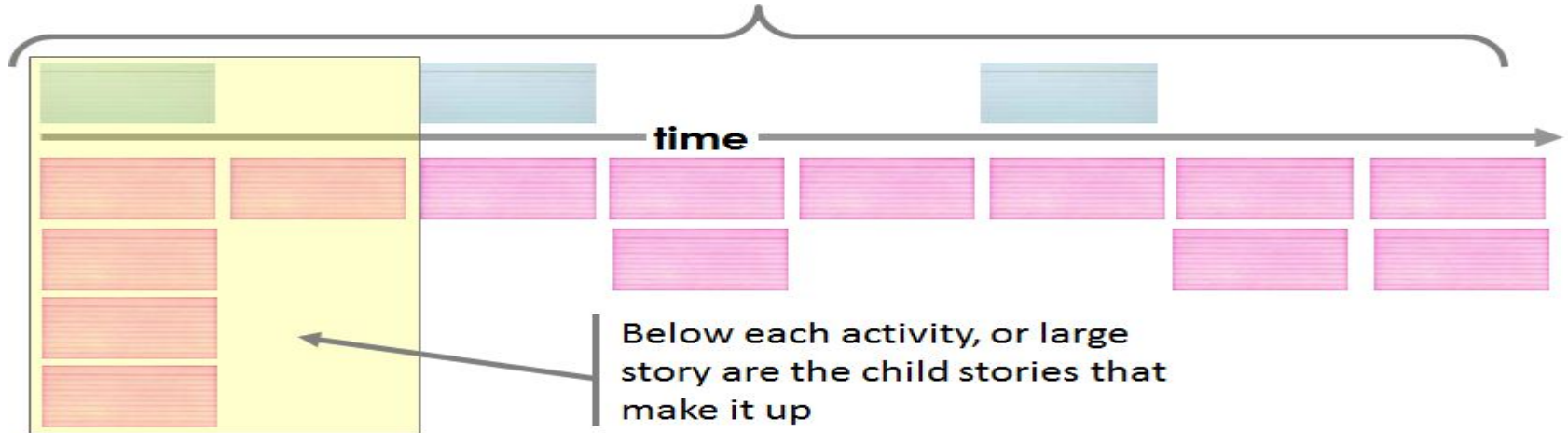
Overlap user tasks vertically if a user may do one of several tasks at approximately the same time  
If in telling the story I say the systems' user typically "does this or this or this, and then does that". "or" signal a stacking vertically, and "then" signal stepping horizontally.



# User story map

The map shows decomposition and typical flow across the entire system.

Reading the activities across the top of the system helps us understand end-to-end use of the system.





## User story map example

TELL A STORY

SEARCH FOR  
ITEM

VIEW PRODUCT

VIEW PHOTO

SELECT ITEM  
FOR PURCHASE

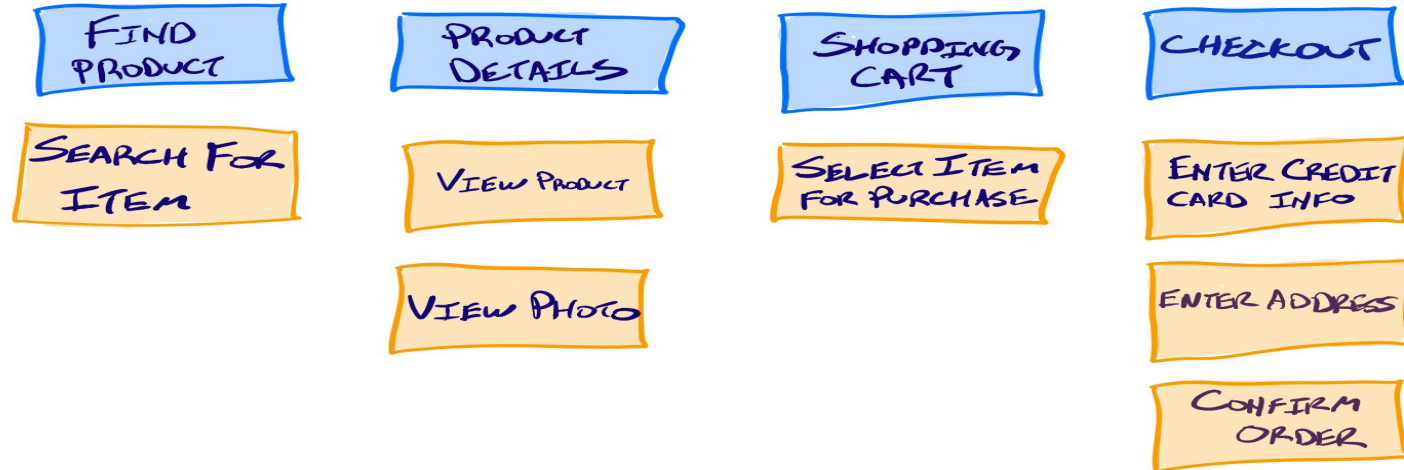
ENTER CREDIT  
CARD INFO

ENTER ADDRESS

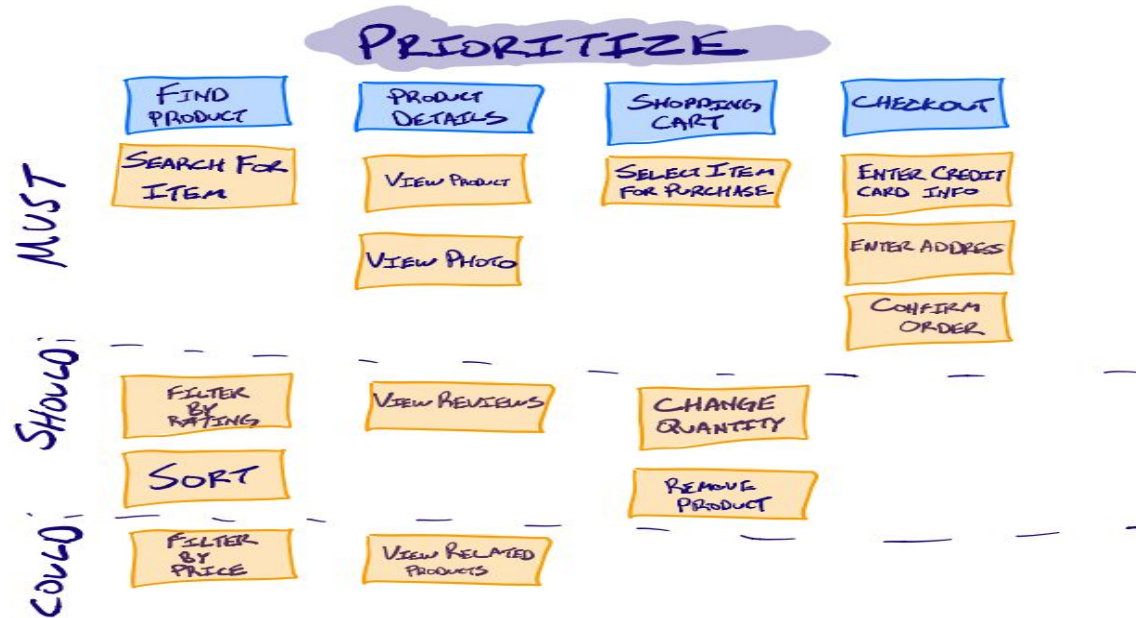
CONFIRM  
ORDER

# User story map example

## GROUP + DEFINE ACTIVITIES



# User story map example



# User story map example

