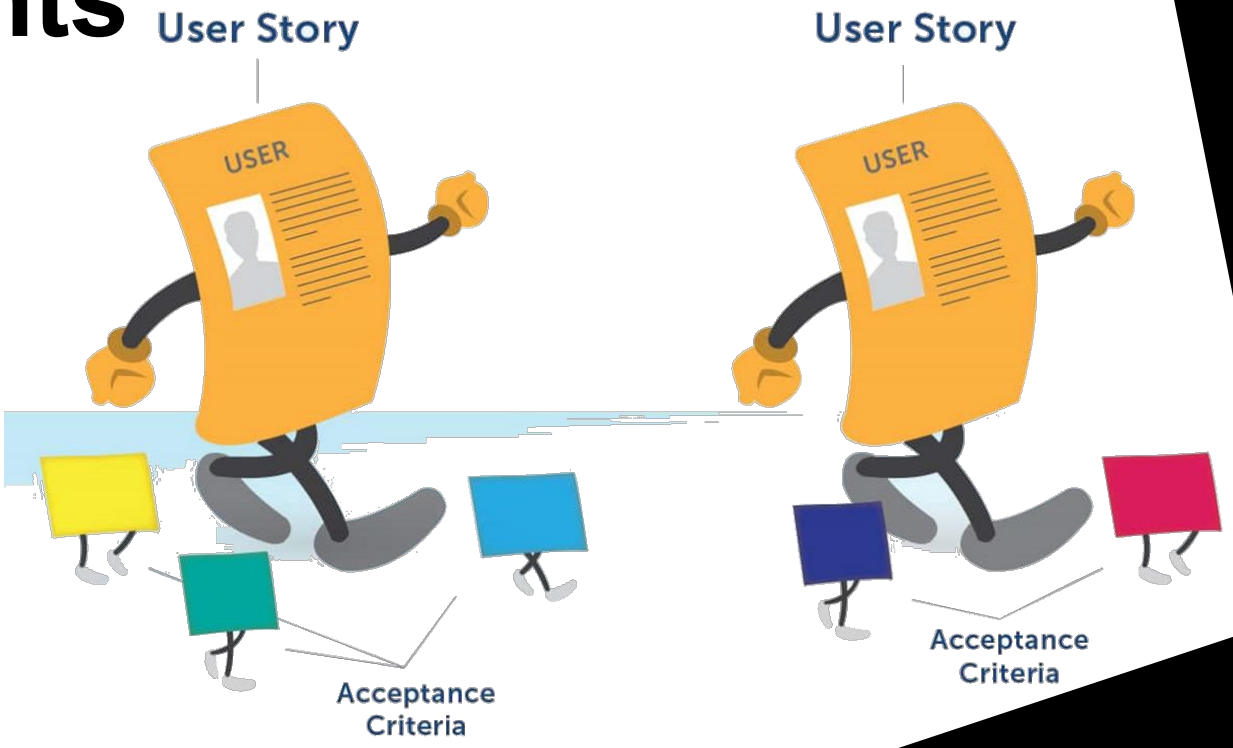
Requirements

Programming 2.2

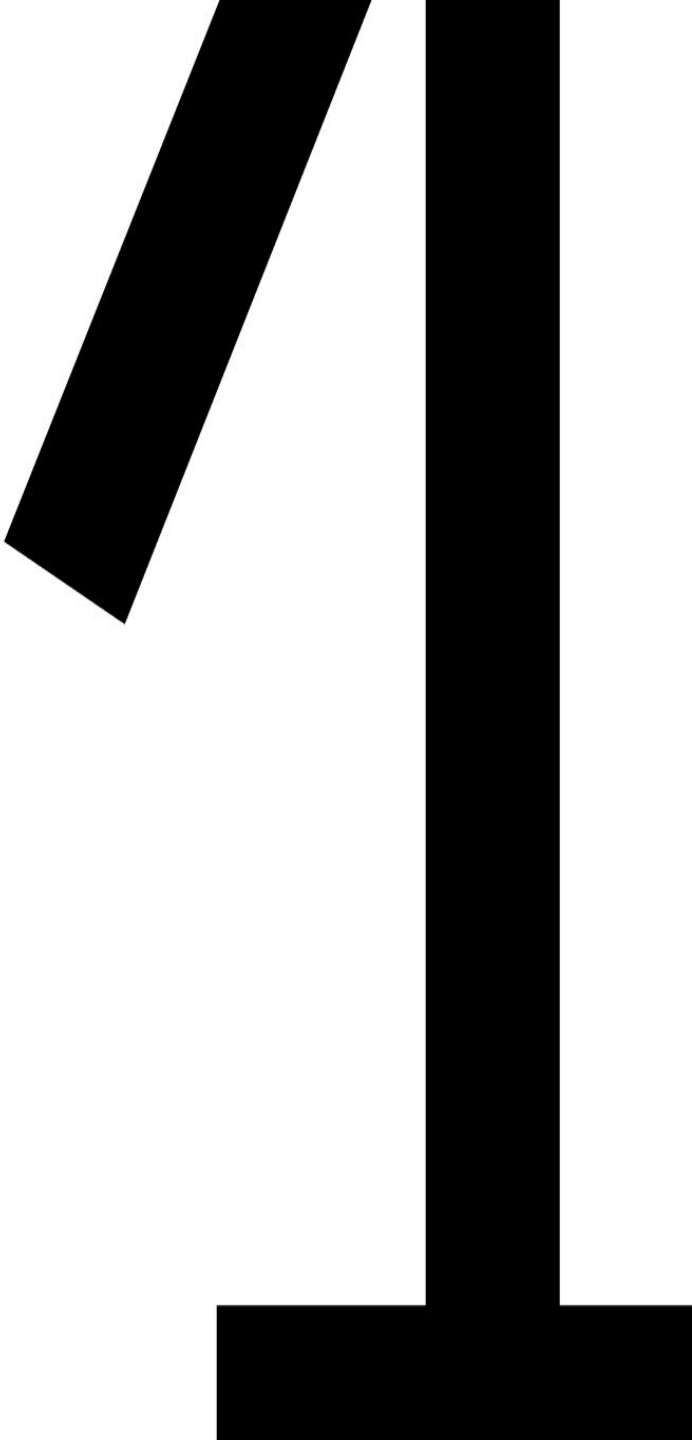
KdG



Agenda

1. User stories
2. Story mapping
3. Acceptance criteria
4. Definition of Ready / Definition of Done

User Stories



Use Cases and User Stories

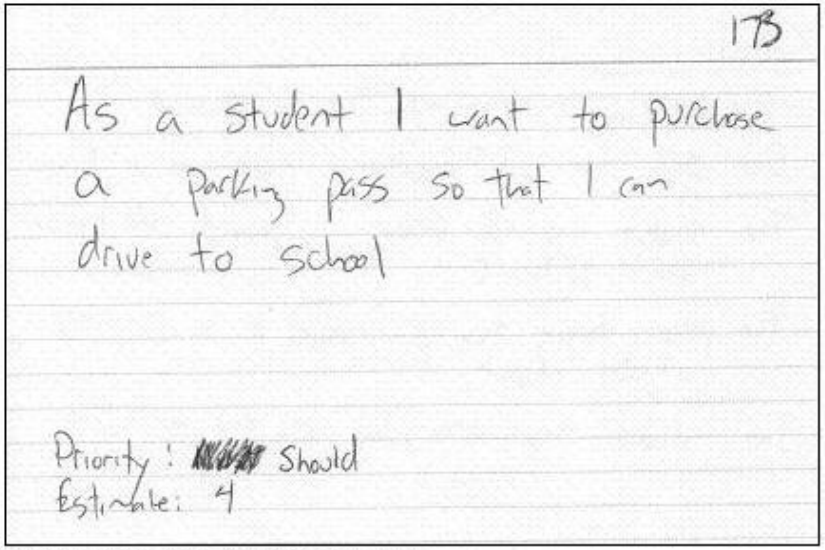
- User stories were discussed in the workshops for the integrated project in the first year
  - please revisit this workshop, it is considered part of this course
- User Story: Pay sale

As a *cashier*
I want to *receive a payment*
So that *a sale can be completed*

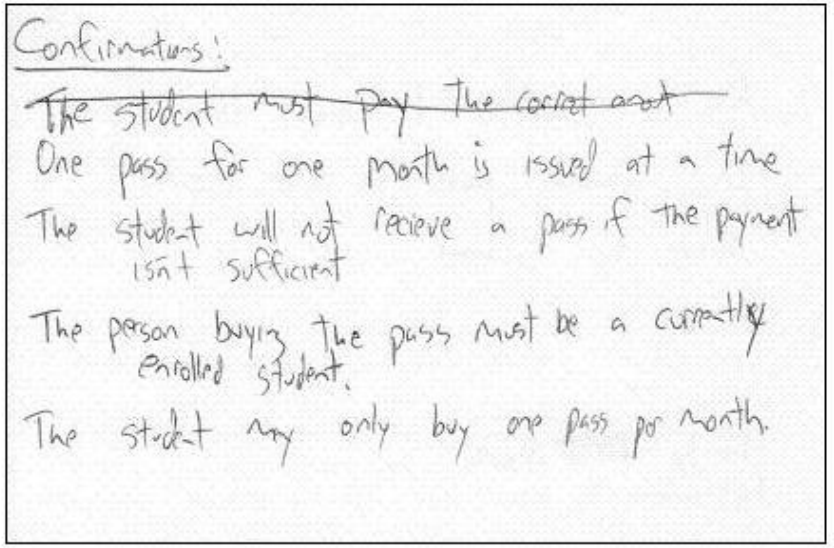
So that: business value

Example: story card

Front of Card



Back of Card



Characteristics of a good user story

- Limit dependencies and make them explicit
- starting point for a discussion. Can be adapted or split up
- **GOAL**: business value
- Small and detailed enough to:
  - estimate
  - Develop in one iteration
  - test

Epic

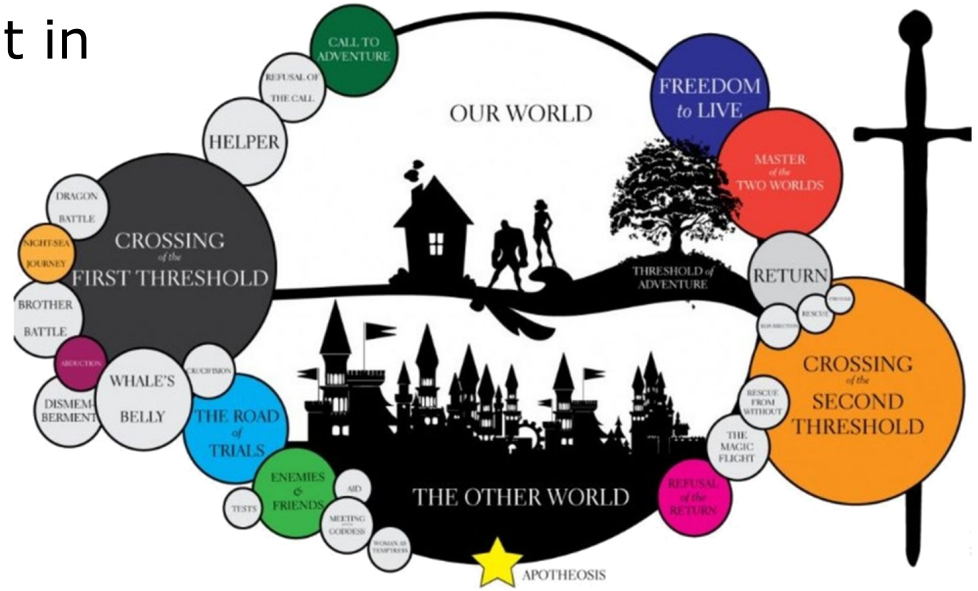
- Adds **hierarchical structure** to stories
- **Epic**: story that can't be built in one sprint
  - Guideline: Split stories > man-week
- Epic: Handle sale

  As *a cashier*

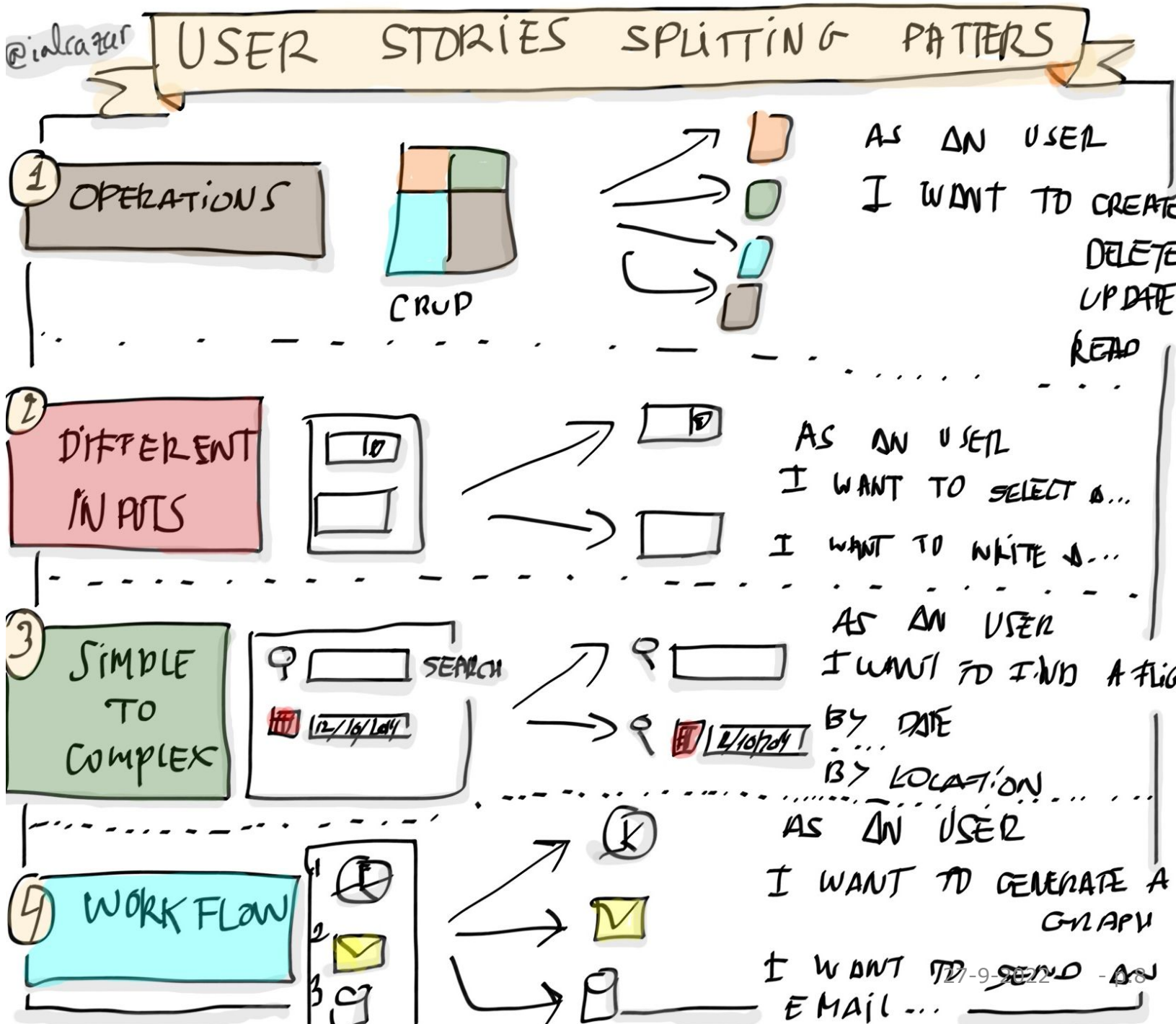
  I want to *handle a sale*

  So that *sales data are stored*

  And *payments are processed*



Epic


@inlcazar
USER STORIES SPLITTING PATTERS
1
OPERATIONS
CRUD
AS AN USER
I WANT TO CREATE
DELETE
UPDATE
READ
2
DIFFERENT INPUTS
ID
AS AN USER
I WANT TO SELECT A...
I WANT TO WRITE A...
3
SIMPLE TO COMPLEX
SEARCH
AS AN USER
I WANT TO FIND A FLIG
BY DATE
BY LOCATION
4
WORK FLOW
AS AN USER
I WANT TO GENERATE A GRAPH
I WANT TO SEND AN EMAIL...

Epic

- Epic *handle sale* contains

1. User story: enter products for sale

As *a cashier*

I want to *enter products*

So that *I know the amount to be charged*

And *there is a record of sold products*

2. User story: pay sale cash

As *a cashier*

I want to *register a cash payment*

So that *I know how much change is due*

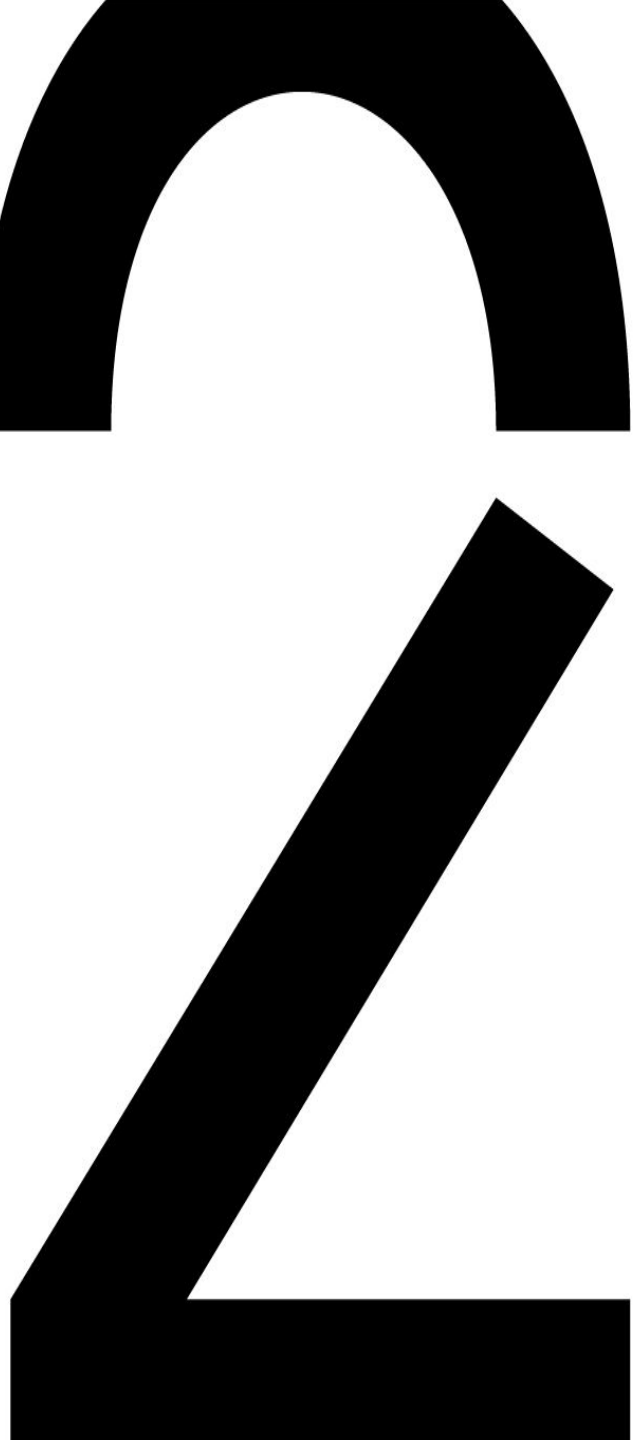
3. User story: pay sale with debit card

As *a cashier*

I want to *handle a debit card payment*

So that *customers can pay without cash*

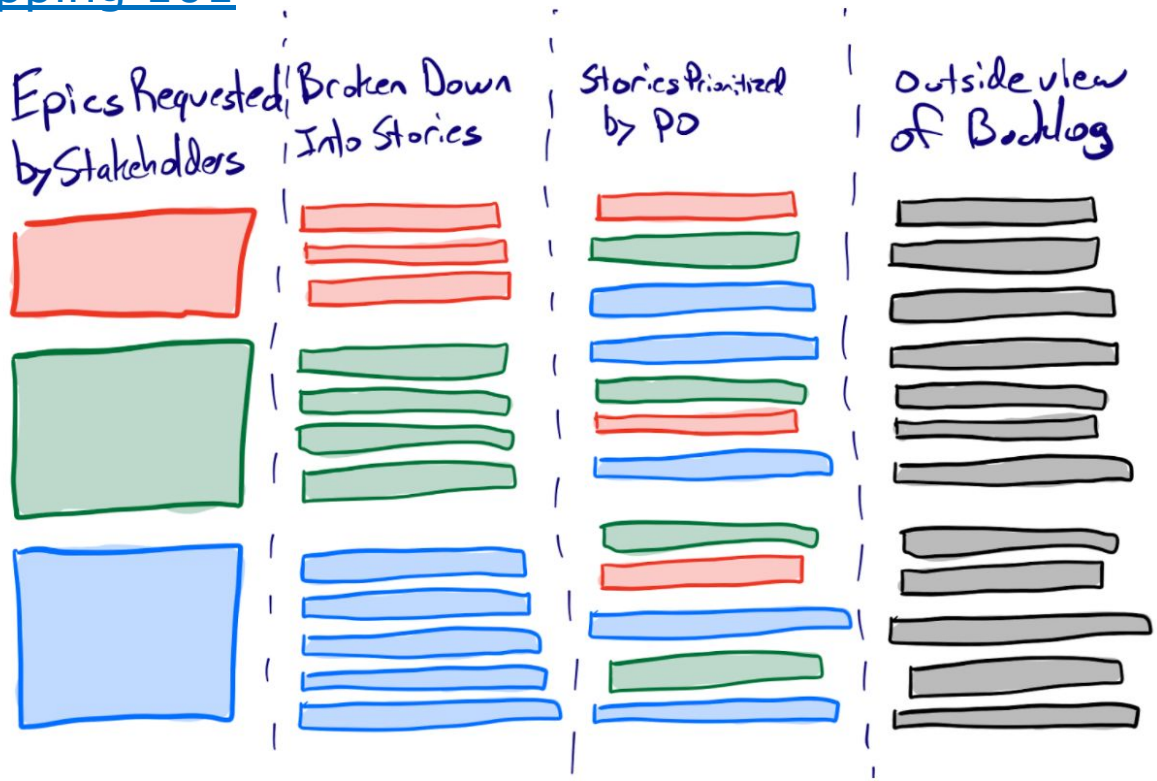
Story mapping



Organise user stories / add on

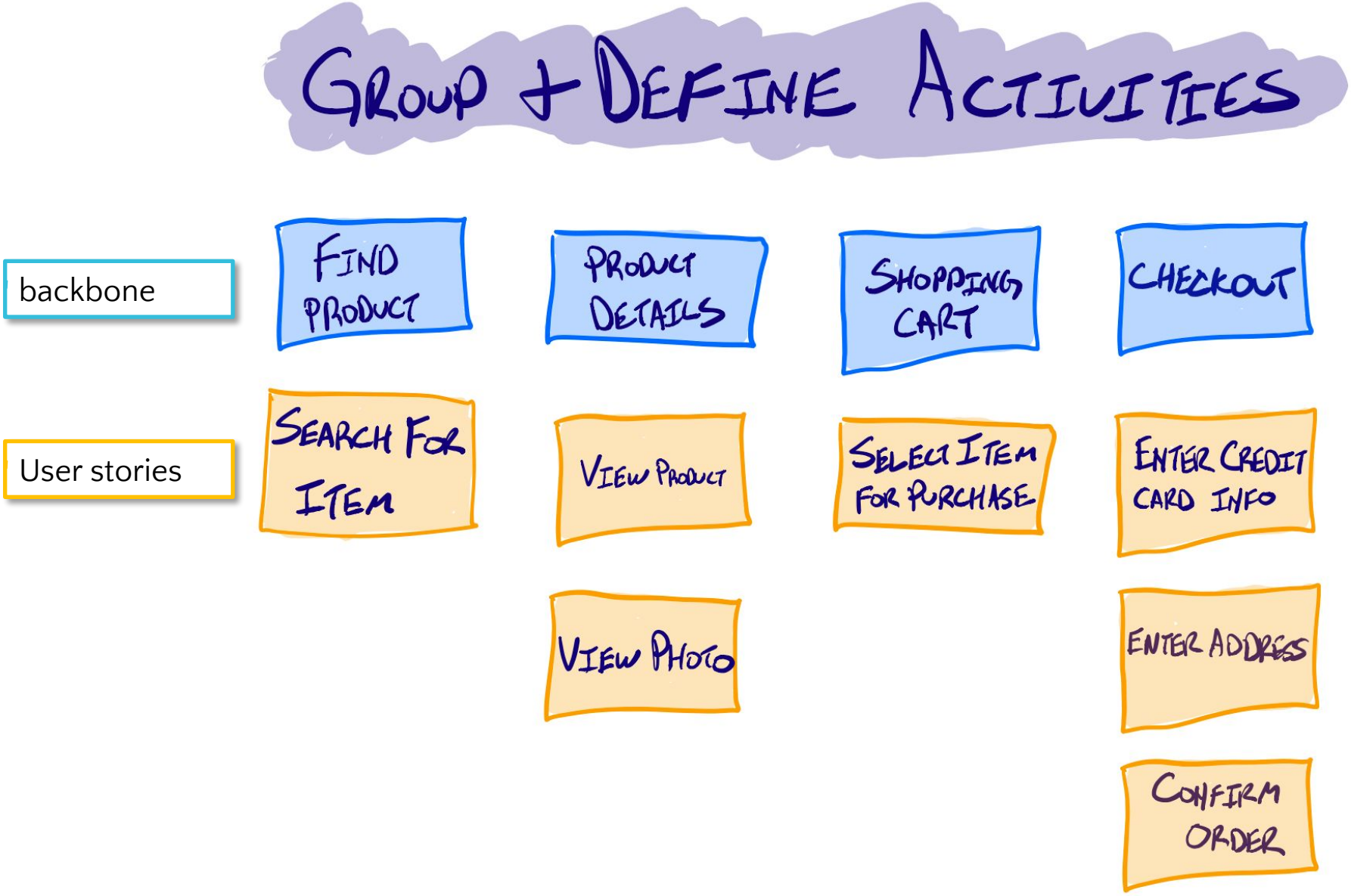
- Add on: user stories are a lightweight technique. Some (but not all) use add-on techniques to manage user stories.
- Problem: product backlog is unstructured and does not show dependencies

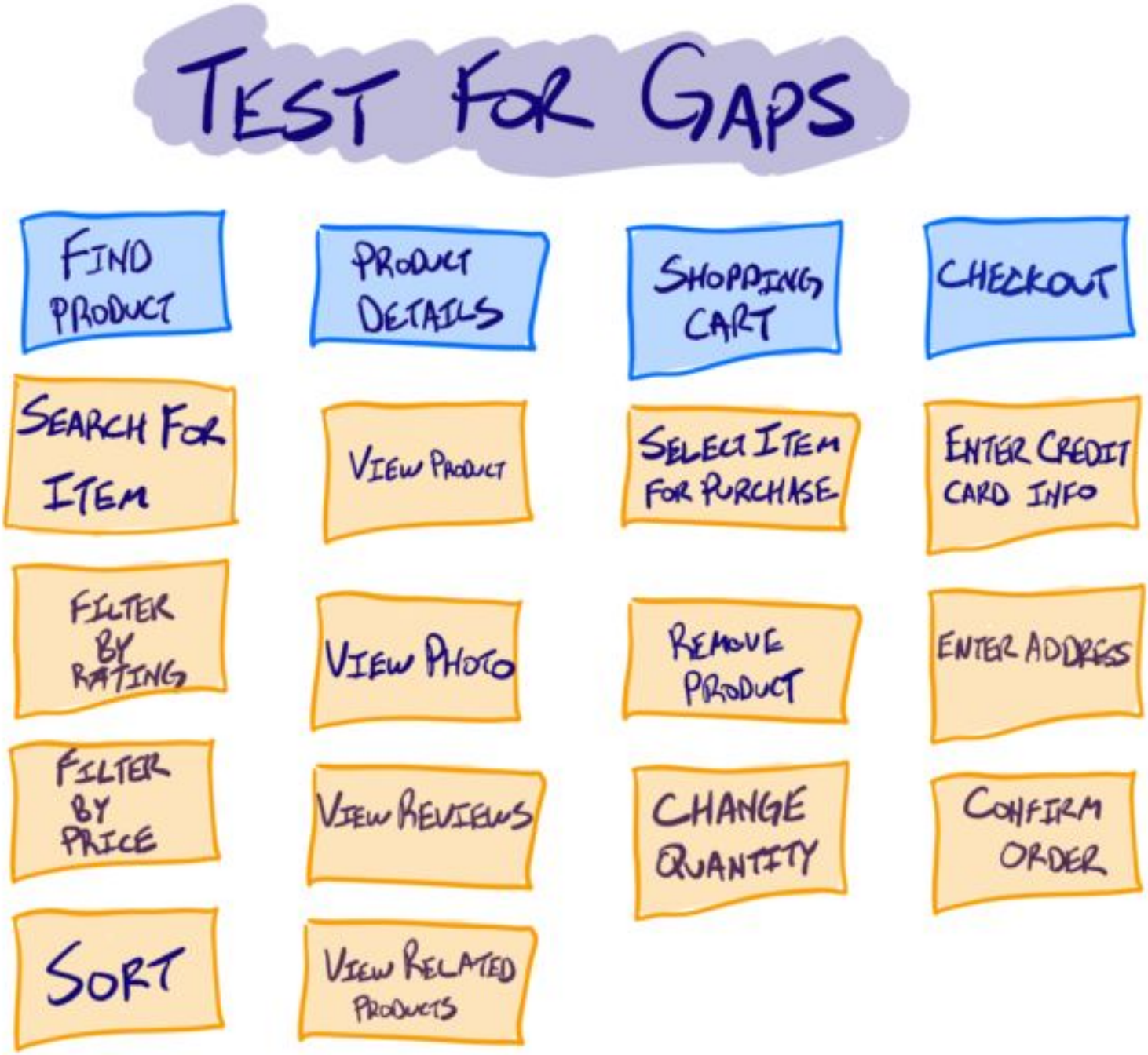
Read [Story Mapping 101]

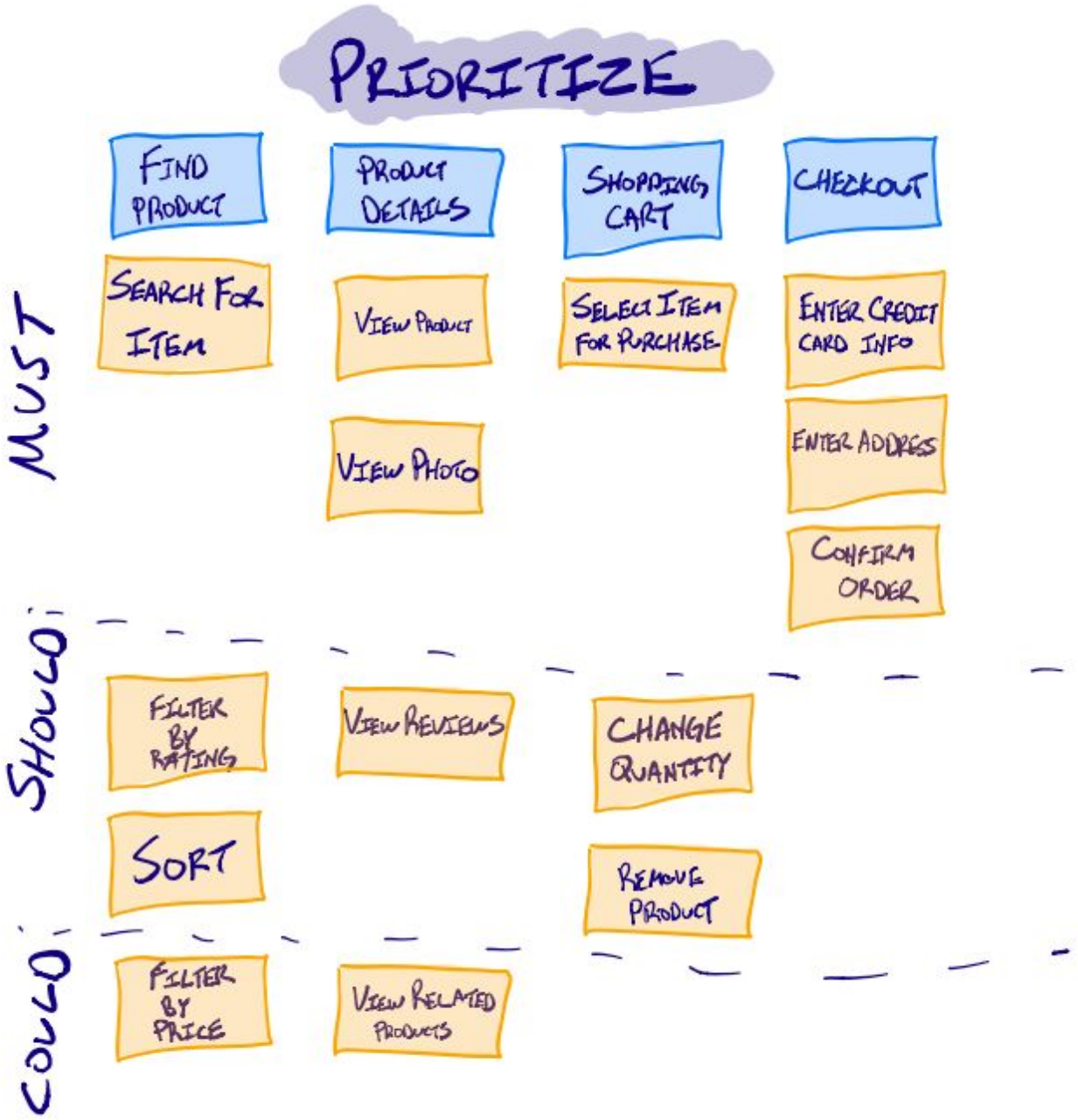


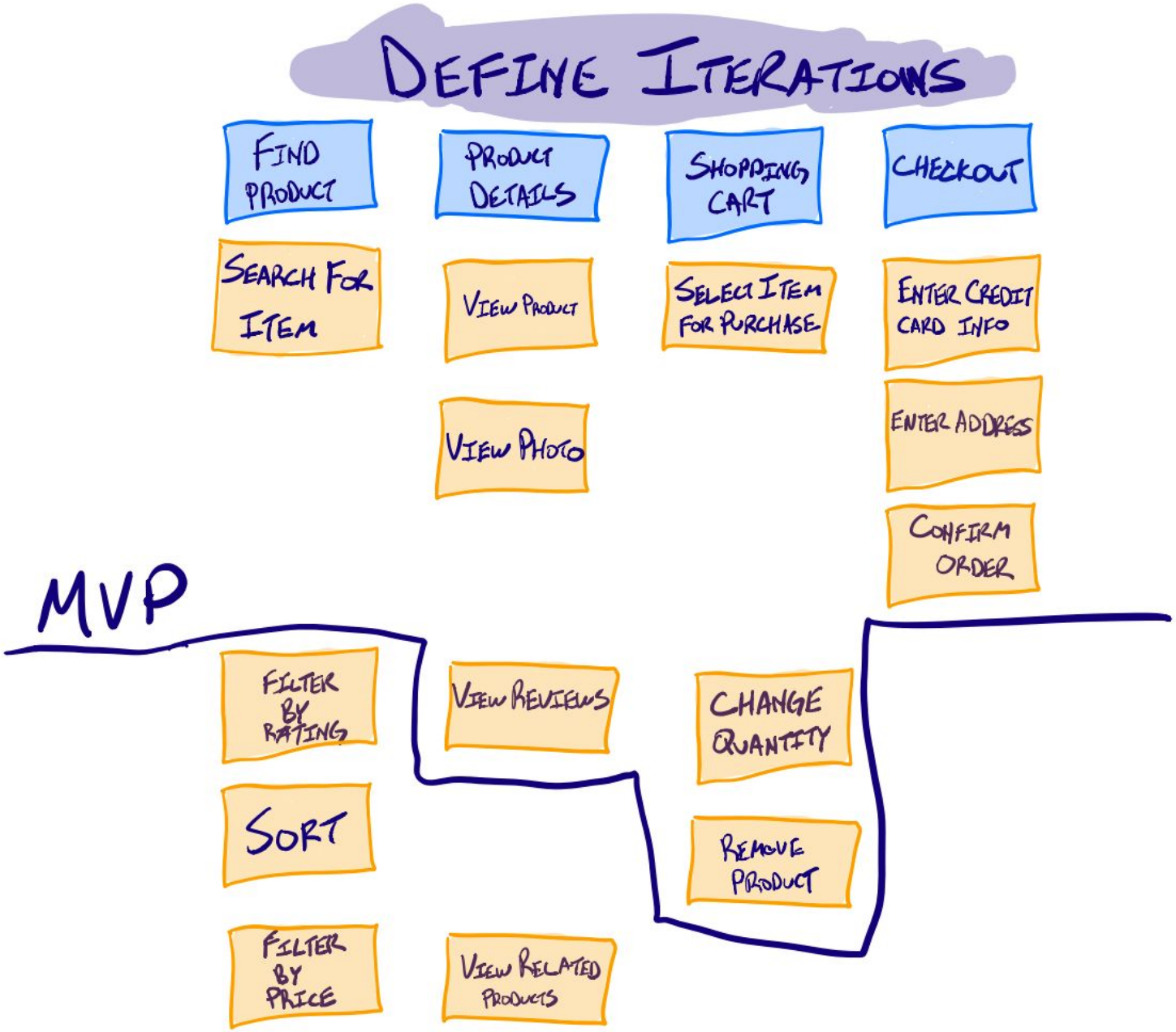

TELL A STORY
SEARCH FOR ITEM
VIEW PRODUCT
VIEW PHOTO
SELECT ITEM FOR PURCHASE
ENTER CREDIT CARD INFO
ENTER ADDRESS
CONFIRM ORDER

Group + Define Activities









Acceptance Criteria

Acceptance Criteria: Scrum add-on

- Behaviour Driven Development is an add-on technique for specifying acceptance criteria for user stories.
  - Add on: not every Scrum process will use this
  - A structured way to add detail to a user story
- With the customer, write scenarios with realistic test data that the realisation of the user story must comply with
  - Include scenarios that go wrong (insufficient payment…)
- Example (Gherkin language) for user story Pay sale:
  - *scenario 1: exact money payment*

current system state

trigger

system action to verify

Given *the sale amount due is €12,34*

When *the customer pays €12,34*

Then *the payment is set to completed*

Acceptance Criteria

- *scenario 2: money payment with change*

  Given *the sale amount due is €12,34*

  When *the customer pays €20*

  Then *the system shows that €7.66 change is to be returned*

  And *the payment is set to completed*

- Typically one scenario tests one behaviour
- Acceptance criteria can include User Experience and non functional requirements (performance). You can express these using Gherkin, but that is out of scope for this course.

Example User Story: enter products

As a cashier

I want to enter products

So that the amount due can be correctly calculated

Example Acceptance Criteria: Enter products

1. **Scenario**: new sale

   Given there is no active sale for register 5
   When 2 items of product "Mars" are entered for register 5
   Then there is an active sale with 1 sales line(s) for cashier 5

2. **Scenario**: active sale

   Given there is an active sale 1235 containing 2 salesline(s)
   for register 4
   When 5 items of product "Twix" are entered for register 4
   Then sale 1235 contains 3 sales line(s)
   And sales line 3 of sale 1235 contains 5 items

3. **Scenario**: close sale

   Given there is an active sale 1235 for register 4
   When the cashier of register 4 closes the sale
   Then sale 1235 is "closed"
   And there is no active sale for register 4

DEFINITION OF READY

The business value and **priority** of the User Story is clear.

The User Story is clear for the TEAM: description and acceptance criteria.

The UX concept is ready (if needed).

Test data and **test devices are ready** (if needed)

All external **dependencies are clear.**

There are no major **open questions** left.

The **User Story** has been **estimated.**

User Story is **small enough** (smaller than one Sprint).

DEFINITION OF DONE

All acceptance criteria met.

UX review is done.

All designs and texts are approved by Editorial, CL and Stakeholders (if needed).

Probe scripts are up to date.

UAT is done done:
- 100% critical/major test cases coverage;
- No Prio 1 and Prio 2 defects open;
- All workarounds are accepted by BA and Stakeholders

Documentation.

KdG
University of Applied Sciences and Arts

Definition of Ready (DoR) / scrum add-on

- criteria a user story must meet to enter a sprint
  - Defined at the start of the project
  - To be fulfilled during backlog grooming
  - Example
    - **I**ndependant: Alle technical and functional elements on which the story depends are available …
    - **N**egotiable: all stakeholders understand and agree on the use case contents
    - **V**aluable: a priority has been assigned to the user story
    - **E**stimable: user story has sufficient detail to estimate effort during sprint planning
    - **S**mall: the user story can be implemented in 1 sprint (preferably by one person)
    - **T**estable: acceptatie criteria are defined, test data are available

Definition of Done (DoD) / scrum add-on

- criteria a user story must meet to be considered completed
- Example:
  - All features are implemented (All TODO's resolved)
  - All acceptance criteria are satisfied
  - Code has been peer reviewed
  - Code has 85% Test Coverage
  - All tests succeed
  - Accepted by stakeholders

Summary

- User stories
- Story mapping
- Acceptance criteria
- Definition of Ready / Definition of Done