



# User Stories – Hoe dan? |

# My experiences on User Stories

- What is a user story and what is it not?
- What does a user story consist of?
- How do you describe these parts of a user story?
- Splitting up user stories, why and how!

# What is a User Story?

- It's a short description of a functional wish
- It's written in a user language
- It always has added value for a user
- It's input for discussion (not a contract)
- Hard to describe! (finding the perfect balance)

# So a User Story is not ....

- A complete functional description.
- System documentation.

# That's the theory ... but HOW (HOE DAN)?

- Decomposition of a User Story?
  - **Description** (who, what, why)
  - **Context** (describes the current situation and the problem/wish)
  - **Acceptance criteria**; to check whether the story matches the expectations

## Students viewing their schedules

**As a student,**  
**I want to** view today's schedule of my lessons,  
**So that** I know when my lessons start and I can be in time.

### Context:

- Currently a student doesn't have any insight in their schedules. They do have an app where they can view their results.

### Acceptatie criteria:

- A student can open an overview of today's lessons in the existing app.
- Per lesson the name, start and endtime and room is shown.
- The next lesson is highlighted.

# User Story - description

- The structure of a ... Who? What? en Why?

- **As a:** (user)

1. Use Persona's.

2. Multiple users -> split-up the story

- **I want to:** (description of what should be delivered)

Use verbs

So not "I want an overview", but "I want to see an overview in the App".

- **So that I:** (describe the reason why this wish is requested)

This can be a hard questions, but mandatory as it determines the VALUE

Is there no good reason 'why' to do this? Then there is no reason to pickup this story.

# Describe WHAT is needed and not HOW

As a student,  
**I want** buttons to filter the search  
results  
**So that** I only see the results that are  
relevant for me



As a student,  
**I want to** be able to find the  
information that is relevant for me  
**So that** I can save time



# Behaviour Driven Design

- A format for the 'Acceptatie criteria'.
  - **Given** — pre-condition for the behavior (can be 1 or more)
  - **When** — the action/event of a user or system
  - **Then** — describe the result (can be 1 or more)
- Example for an ATM:
  - **Given:** a debit card is activated; AND the balance of the account is  $> \text{€}0$
  - **When:** the customer wants to withdraw money from his account
  - **Then:** the balance is lowered; the ATM will dispense the money; the receipt is printed; the card is returned.



# Having this all applied? ... do the INVEST-check (all members)

## Independent?

Make sure your user stories is independent from other stories, this enables the product owner to set priorities.

## Negotiable?

User stories shouldn't be too detailed. Developers should have freedom to find the best solution.

## Valuable?

The functionality should add value. This value is determined by the product owner.

## Estimable?

Do you have enough information? Can it be estimated how long (relatively) it will take to deliver this story?

## Small enough?

Is the story small enough? Don't make it too small as the 'value' will also be small.

## Testable?

Have the right acceptance criteria been added? And are they SMART enough? Can they be verified?

# Splitting User Stories

- Add business value asap.
- Avoid splitting bij architectural layer (like Database, Interfaces).
- Use 1 or more of the splitting patterns

1. Workflow steps

2. Business Rule Variations

9. Break out a Spike

3. Major effort

8. Operations (CRUD)

**Splitting user stories**

4. Simple/Complex

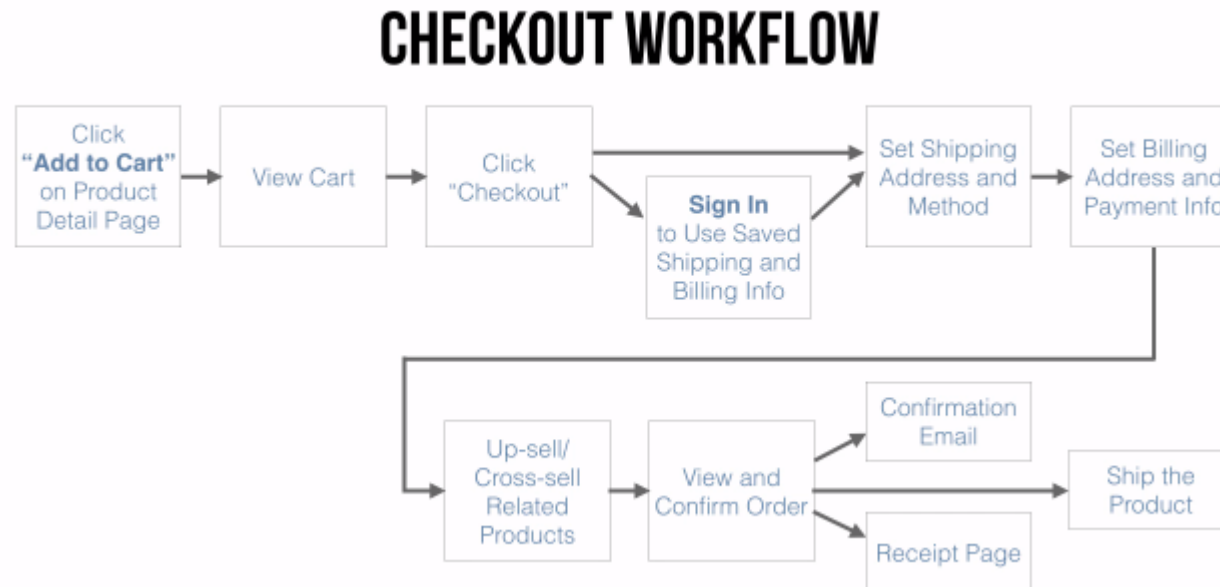
7. Defer Performance

5. Variations in Data

6. Data entry methods

# #1 Workflow steps

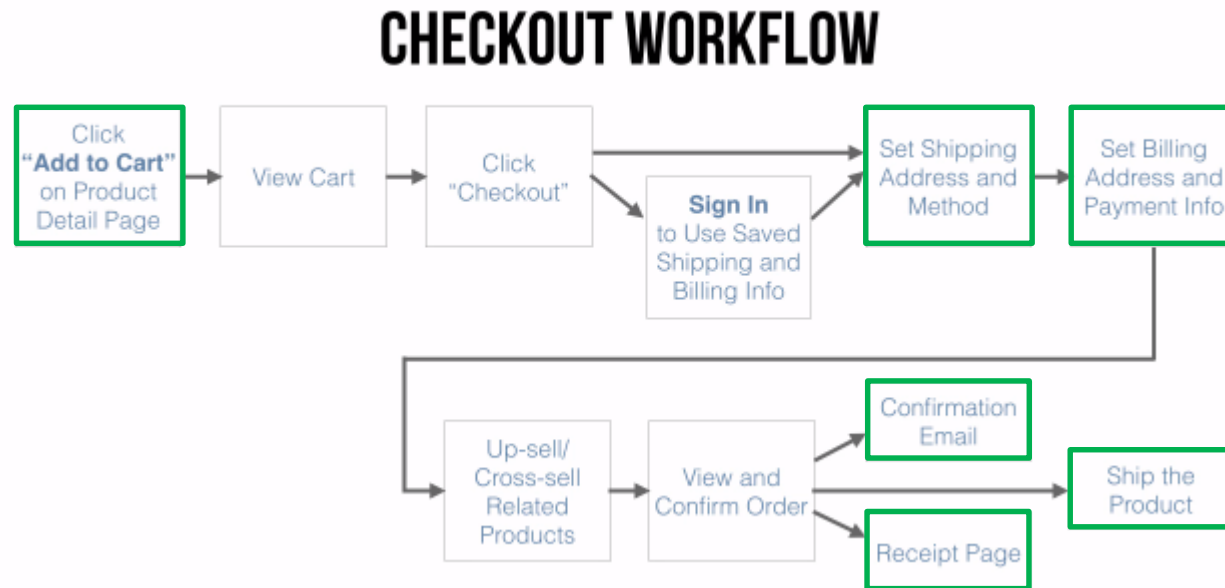
- Identify the most important steps:
  - START and END of a flow is most valuable
  - Do not split from LEFT to RIGHT



Animated slide  
#1

# #1 Workflow steps

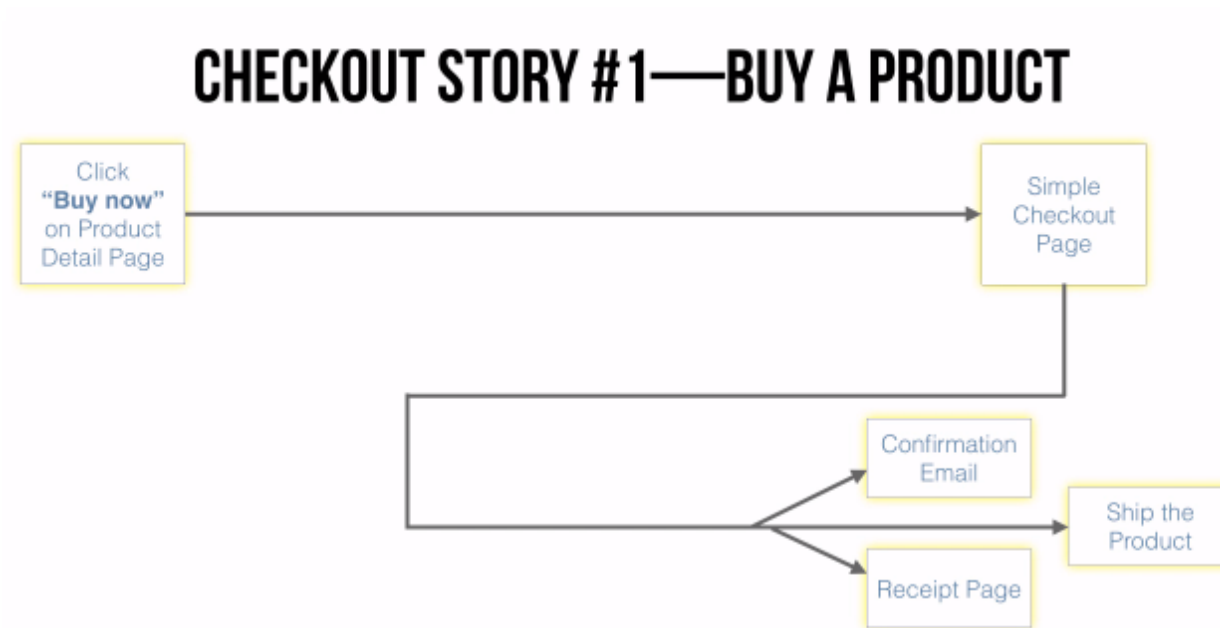
- Identify the most important steps:
  - START and END of a flow is most valuable
  - Do not split from LEFT to RIGHT



Animated slide  
#2

# #1 Workflow steps

- Identify the most important steps:
  - START and END of a flow is most valuable
  - Do not split from LEFT to RIGHT



Animated slide  
#3

## #2 Business Rule Variations

- Can business rules be split?
- Or should it be split as the business rule is not SMART enough?

**| As a user, I can search for flights with flexible dates.**

Can be split into:

**| ...as "between date x and y."  
| ...as "a weekend in December."  
| ...as " $\pm$  n days of date x."|**

## #3 Major effort

- Identify substantial effort but ...
- Keep stories independent!

**| As a user, I can pay for my flight with iDeal, AfterPay, PayPal.**

Can be split into:

**| ...I can pay with one type (or iDeal, AfterPay, PayPal).**  
**| ...I can pay with all 3 types (iDeal, AfterPay, PayPal).**  
*(given one type is already implemented)*

## #4 Simple / Complex

- When during refinement stories are getting larger and larger -> Ask yourself "What's the simplest version of this?"

**| As a user, I can search for flights between two destinations.**

Can be split into:

**| ...specifying a max number of stops.  
...including nearby airports.  
...using flexible dates.  
...etc....**



## #5 Variations in Data

- Complexity in a story can come from handling variations in data.
- Data can be anything like: location, language, role, type of customer.

**| As a customer, I can see all my telecom product that I ordered.**

Can be split into:

**| As a residential customer, I can see ....**  
**| As a soho customer, I can see ....**  
**| As a enterprise customer, I can see ....**

## #6 Data Entry Methods

- Complexity sometimes is in the user interface rather than in the functionality itself.

**| As a user, I can search for flights between two destinations.**

Can be split into:

**| ...using simple date input.**  
**| ...with a fancy calendar UI.**

## #7 Defer performance

- In Agile, priority is to add business value and learn.
- If necessary, learn from slow implementation

**| As a user, I can search for flights between two destinations.**

Can be split into:

**| ...(slow—just get it done, show a “searching” animation).**  
**| ...(within 5 seconds).**

## #8 Operations (e.g. CRUD)

- The word 'manage' covers multiple operations.
- Identify the various operations and prioritize them.

**| As a user, I can manage my account.**

Can be split into:

**| ...I can sign up for an account.**  
**| ...I can edit my account settings.**  
**| ...I can cancel my account.**

## #9 Break Out a Spike

- When business part is unclear.
- When implementation is poorly understood

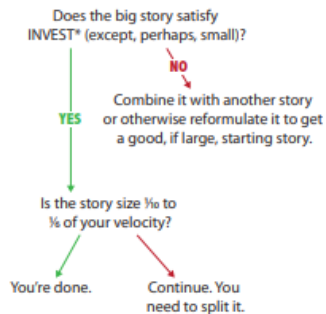
**| As a user, I can pay by credit card.**

Can be split into:

**| Investigate credit card processing.**  
**| Implement credit card processing.**

# HOW TO SPLIT A USER STORY

## 1 PREPARE THE INPUT STORY



\* INVEST - Stories should be:  
Independent  
Negotiable  
Valuable  
Estimable  
Small  
Testable



AGILEFORALL  
www.agileforall.com

## WORKFLOW STEPS

Can you split the story so you do the beginning and end of the workflow first and enhance with stories from the middle of the workflow?

Can you take a thin slice through the workflow first and enhance it with more stories later?

## DEFER PERFORMANCE

Could you split the story to just make it work first and then enhance it to satisfy the non-functional requirement?

Does the story get much of its complexity from satisfying non-functional requirements like performance?

## SIMPLE/COMPLEX

Could you split the story to do that simple core first and enhance it with later stories?

Does the story have a simple core that provides most of the value and/or learning?

Could you group the later stories and defer the decision about which story comes first?

## MAJOR EFFORT

Can you split the story to handle data from one interface first and enhance with the others later?

## INTERFACE VARIATIONS

Is there a simple version you could do first?

## BREAK OUT A SPIKE

Are you still baffled about how to split the story?

Can you find a small piece you understand well enough to start?

Write that story first, build it, and start again at the top of this process.

Can you define the 1-3 questions most holding you back?

Write a spike with those questions, do the minimum to answer them, and start again at the top of this process.

Take a break and try again.

## 2 APPLY THE SPLITTING PATTERNS

## OPERATIONS

Can you split the operations into separate stories?

Does the story include multiple operations? (e.g. is it about "managing" or "configuring" something?)

## BUSINESS RULE VARIATIONS

Can you split the story so you do a subset of the rules first and enhance with additional rules later?

Does the story have a variety of business rules? (e.g. is there a domain term in the story like "flexible dates" that suggests several variations?)

## VARIATIONS IN DATA

Does the story do the same thing to different kinds of data?

Can you split the story to process one kind of data first and enhance with the other kinds later?

## BREAK OUT A SPIKE

Are you still baffled about how to split the story?

Can you find a small piece you understand well enough to start?

Write that story first, build it, and start again at the top of this process.

Can you define the 1-3 questions most holding you back?

Write a spike with those questions, do the minimum to answer them, and start again at the top of this process.

Take a break and try again.

Visit <http://www.richardlawrence.info/splitting-user-stories/> for more info on the story splitting patterns  
Copyright © 2011-2018 Agile For All. All rights reserved.

Last updated 2/21/2018

# Vragen



**Bewaar ze voor de Lean Coffee**