Essence
Software Engineering Essentialized
Part 3B – Running with User Story Lite

Giuseppe Calavaro, Ph.D.

Agenda of this Teaching Module

- User Stories Introduction
  - How to write good user stories: INVEST
- User Stories Lite Practice using Essence example
  - User Stories Lite Alphas
  - User Stories Lite Products
  - User Stories Lite Activities
  - Splitting User Stories Pattern

Example of using User Stories Lite Practice on our project are in green boxes

- The Value of the Kernel to the User Story Practice
- Impact of User Stories Lite practice for the team

Introduction to User Story concept

- The User Story practice is a popular practice, in particular for small teams.
  - It originated from Extreme Programming (XP), a lightweight, efficient, low-risk way to develop software.
  - User stories have the benefit of getting the team to think, enquire and understand the value of what they do from the point of view of their users.
- The User Story Lite practice is a simplified version of the User Story practice
  - It was created just for the readers of this book to understand how to Essentialize a practice and how to use an Essentialized practice

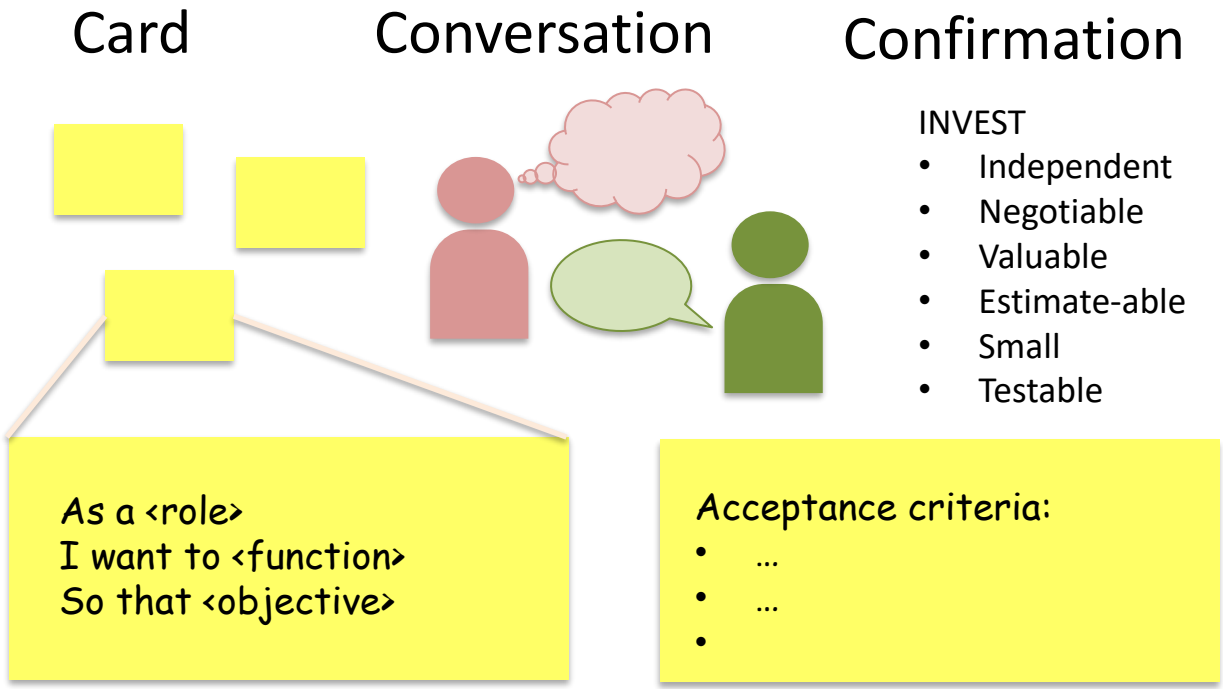
User Stories Explained

- A user story describes functionality in the system we are building that is valuable to a user of a system.
- A user story includes a written description that is used when discussing the story along with tests to help communicate what is needed to complete the story.
  - The idea of user stories is to provide a way to facilitate discussion to help clarify who (i.e. a role) a piece of functionality is for and how it benefits the role.

Capturing User Stories

- A user story is often captured on a 5 x 3 index card with a very concise format
  - As a <role, or type of user>,
  - I want to <list here the function you want the system to do>,
  - so that <list here the objective you want to achieve>
- This template helps to ensure that the "Who", "What" and "Why" are all considered and captured.
- An example could be:
  - As a bank customer
  - I want to have a direct deposit capability
  - so that my employer can electronically send me my paycheck.
- Who – will get the value?
- What – do we need to achieve?
- Why – are we doing it?

User Story Lite Practice Big Picture



- **Card** – a succinct headline description, as captured on a Story Card
- **Conversation** – the actual users of the proposed system and developers discuss what is needed to converge on the best solution
- **Confirmation** – acceptance criteria, captured as bullet-point statements, which can be captured on the back of the Story Card.

How to write good User Stories: INVEST

1. **Independent** - *independent* of each other
   - So that each can be developed separately.
2. **Negotiable** – written to promote negotiation between the user and developers
   - Negotiation promotes understanding and commitment.
3. **Valuable** - A user story should be *valuable* to the user.
   - To understand the real intent of a requirement
4. **Estimatable** - enough details to allow the developer to estimate the work effort required to implement the story
5. **Small** - to fit within a given iteration
   - Stories that are too large to fit within an iteration are referred to as Epics.
6. **Testable** - when completed it has to be *testable*.
   - Writing the tests first help ensure the story is testable and helps ensure both users and developers are in agreement on what it means to complete the story.

Why do we need the "so that" clause?

- To ensure developers **understand the end objective of the user**
- This helps to support evolutionary requirements development.
  - By evolutionary requirements development we mean that the requirements may evolve as we learn more about the available options and needs of the user.
- This also keeps the developer's options open in providing alternative solutions.

User Stories Lite Essentialized Practice

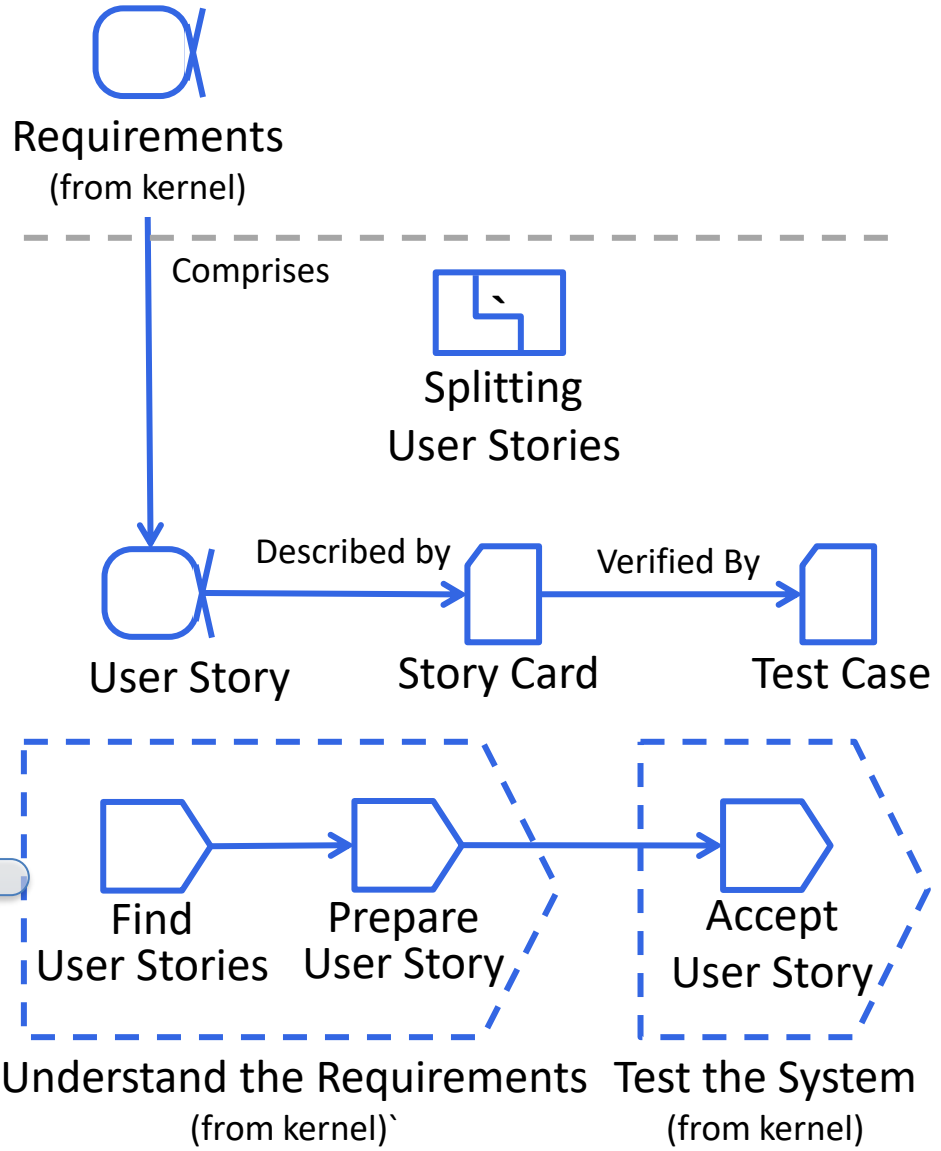
- For a deep dive on User Stories other literatures provide complete and detailed presentations
- Our **goal is to show how to essentialize a Practice**
  - This is a Lite practice because we have selected what we deem as a minimal core of the practice;
    - minimal for demonstration but in the real world would require much more technical detail

- How to describe the User Stories Lite Practice using Essence?
  - The first questions we always ask when essentializing a practice are:
    - What are the things you need to work with?
    - What are the activities you do?

User Stories Lite Practice expressed in Essence

- **Requirements** are decomposed into **User Stories**
- Each User Story is:
  - described by a ***Story Card***
  - Verified by a ***Test Case***
  - Singleton work products for each User Story
- The practice has several activities:
  1. Find User Stories
  2. Prepare User Story
  3. Accept User Story
- A pattern ***Splitting User Stories*** represent the approach to help teams split user stories to ease development

User Story Lite practice is simpler than Scrum Lite. Not only User Story Lite has fewer elements than Scrum Lite, it also relates to fewer elements in the kernel.



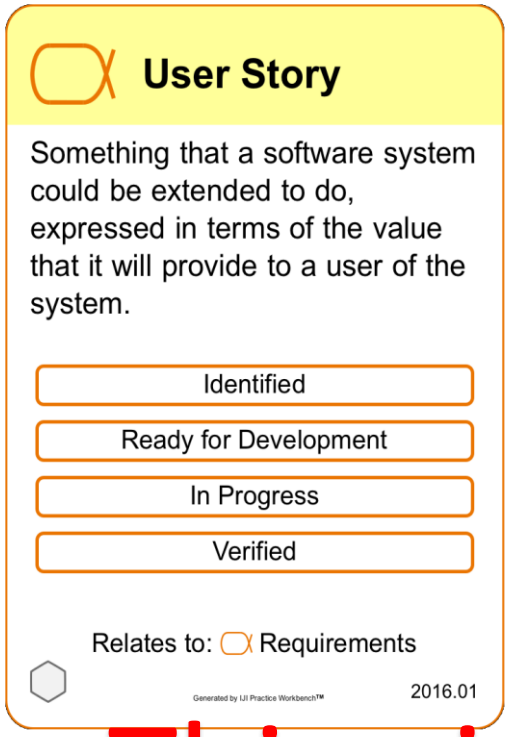
Elements of the User Story Lite Practice

Element	Type	Description
User Story	Alpha	Something that a software system could be extended to do, expressed in terms of the value that it will provide to a user of the system
Story Card	Work Product	An index card, or equivalent, that captures the essential details of a user story
Test Case	Work Product	Defines test inputs and expected results to evaluate whether a user story is fully and correctly implemented
Find User Stories	Activity	Identify things of value that a software system could do. Capture these as simple and succinct headline descriptions on story cards

Element	Type	Description
Prepare a User Story	Activity	A user story is prepared for development by discussion with users to build understanding and refinement of its acceptance criteria and test cases
Accept a User Story	Activity	The user story implementation is evolved in close collaboration with the customer/user until it is acceptable to and accepted by the customer/user representative.
Splitting User Stories	Activity	Small things get done faster. In agile development there is a continuous and relentless drive to reduce the size of user stories by splitting bigger stories into smaller ones. The key is to ensure that each story delivers value.

User Story Alpha

- A User Story is something that a software system could be do expressed in terms of the value that it will provide to a user of the software system



User story progresses through the following states:

- **Identified** – The user story is identified with its value clearly expressed. It is placed in the team's product backlog.
- **Ready for Development** – The team discusses the details of the user story such that members are clear on what is involved in fulfilling the requirements behind the user story This might involve details about user interfaces, implementation details, and so on.
- **In Progress** – At this state, the team is working on fulfilling the user story.
- **Verified** – The user story implementation is verified by a qualified user representative, such as a member being the Product Owner

This picture is snapshot

User Story Alpha States

- We are missing the cards for each state of the
- User Story Alpha

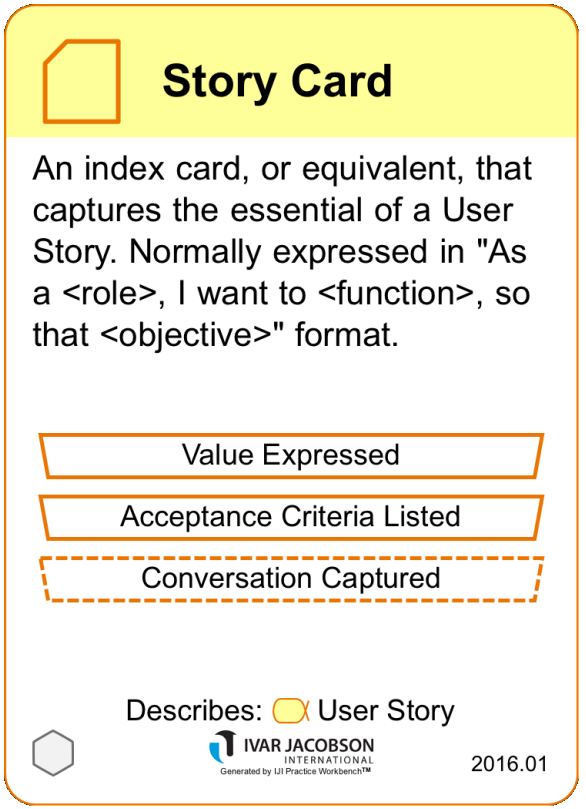
- The alpha state cards above help understand and agree when the Alpha has achieved a state with specific checklists

Story Card Product

A Story Card is the written description of a user story

A user story can be at different levels of detail:

- **Value Expressed** – The value of the user story is clearly expressed, such as using the common format described before (and in the card)
- **Acceptance Criteria Listed** – The acceptance criteria for the fulfilment of the user story are clearly expressed
- **Conversation Captured** – The discussions the team has about the user story are captured so that the team understands more clearly the requirements for the user story and the rationale behind the details of the user story.
  - These discussions are usually verbal, but can be written on the story card itself or some electronic means



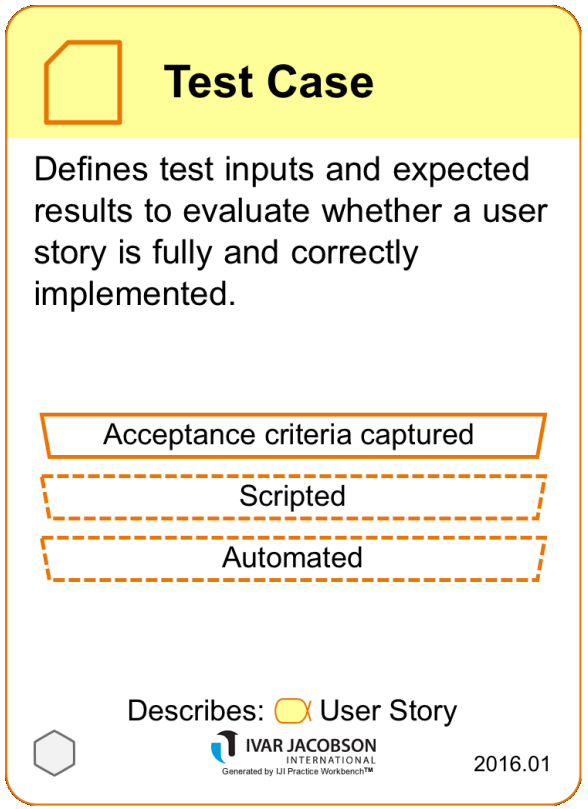
This picture is snapshot

Test Case Product

A Test Case defines test inputs and expected results to evaluate whether a user story is fully and correctly implemented.

A test case has several levels of detail:

- **Acceptance criteria captured** – The different possible ways for testing the user story are captured.
- **Scripted** – The step-by-step procedure for testing and accepting the user story is available. This necessitates also the preparation of test data and test environment used when executing the test case.
- **Automated** – The test case is automated and can be executed with little or no intervention.



This picture is snapshot

Kick Starting User Story Lite practice

There were two primary challenges our development team faced that led them to decide to try User Stories Lite on their endeavour.

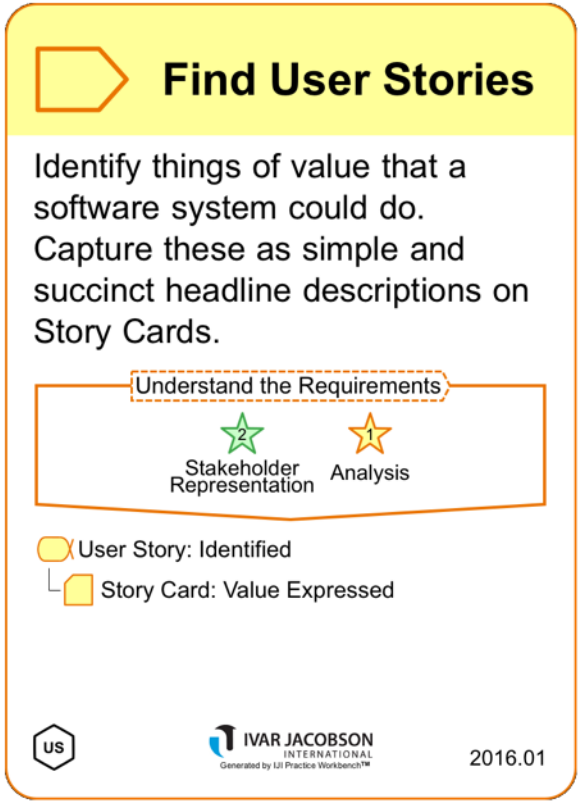
1. Smith's team members sometimes found themselves wondering about the purpose of the system they were developing.
   - Instead of just enumerating product backlog items, developing product backlog items in a user story form the resulting requirements would help the team better understand the purpose of the system they were developing.
   - This would also help Angela (the Product Owner) when discussing the system with her stakeholders, such as Dave.
2. The second challenge the development team often faced was product backlog items being too large to fit within a single sprint/iteration.

Working with user stories involves several activities

- First the team needs to **find user stories**, **prepare each user story** for development, and then **accept the implementation of the user story**.
- The implementation of the user story, (i.e. writing and testing code), is outside the scope of the User Story Lite practice we are describing
  - It is expected to be addressed by another practice such as microservice practice presented later

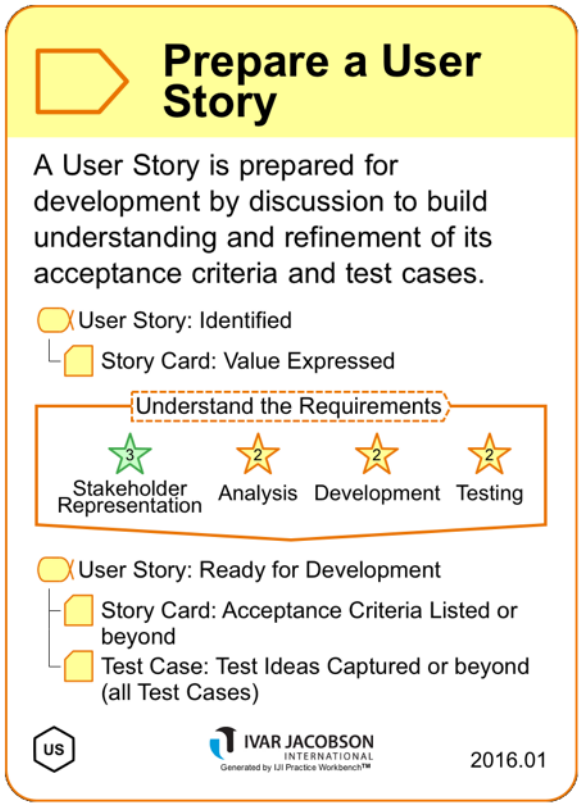
Find User Stories Activity

- The Find User Stories activity is about agreeing on the how the users are going to use the Software System for something that brings value to them.
  - For each User Story found a Story card is created with a simple headline and the value expressed
- The card indicates that:
  - The Story Card needs at a minimum to achieve the **Value Expressed** level of detail
  - The user story alpha needs to achieve the **Identified** state



Prepare a User Story Activity

- The Prepare a User Story activity is about discussing how the user will interact with the system
  - The discussion aim to achieve a better understanding of the system to develop, estimate the effort to implement and describe the acceptance criteria
- The card indicates that:
  - The user story alpha needs to achieve the **Ready for Development** state by having
    - The Story Card needs at a minimum to achieve the **Acceptance Criteria Listed** level of detail
    - The Test Case needs at a minimum to achieve the **Test Ideas Captured** level of detail



Travel Essence performing the practice

Angela and the team were discussing which product backlog items they would target for the next iteration. They selected the following three backlog-items:

1. Improve algorithms to rank destinations according to traveller specific preference
2. Improve algorithms to rank destinations according to general popularity of destinations
3. Collect user data from users and analyze them

Together as a team they expressed the user stories as shown in Figure

As a traveller, I want to have destinations I like to be ranked higher than other destinations so that it is easier for me find these destinations

Acceptance criteria:

1. A visited destinations ranks higher than a non-visited one
2. A "liked" destination ranks higher than a "non-liked" destination

As a traveller, I want to have popular destinations ranked higher than other destinations so that it is easier for me find these destinations

Acceptance criteria:

1. Each destination visited by a traveller will be given a higher score
2. Each destination liked by a traveller will be given a higher score

As a TravelEssence promotion staff, I want to track the actions on the recommendation list so that I can improve the quality of the recommendation and user experience.

Acceptance criteria:

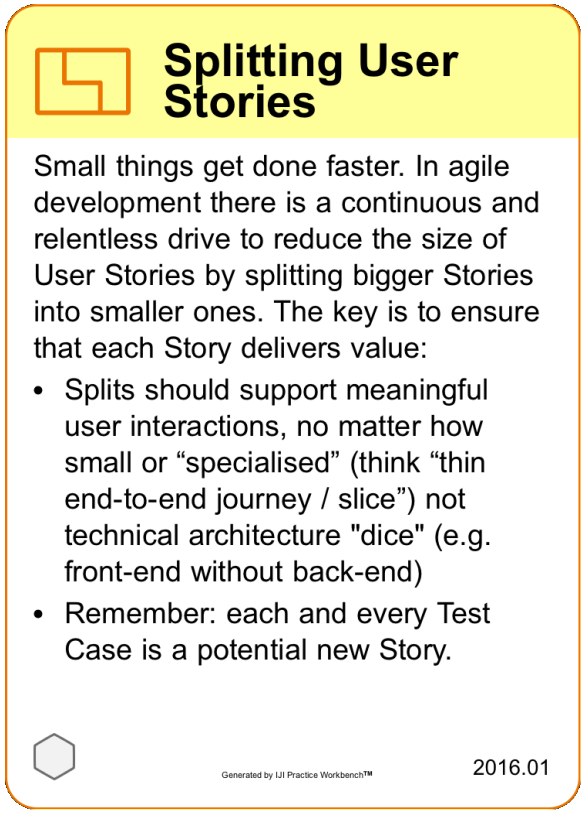
1. Count the clicks, likes and booking on each recommendation destination by specific traveller and travellers in general.
2. Trend chart by day, week, month of top N destinations

What is the outcome for the team?

- Tom, Joel and Grace were much happier with the user story form as depicted in the Figure as this form helped them better understand the purpose of the system they were developing.
  - The added detail helped them estimate each story and ensure each story was small enough to fit in a iteration.
  - In the first and second user stories starting from the left in Figure, because we now have agreed to acceptance criteria the team better understands what Angela will accept for improved algorithms.
  - In the third user story because it specifies "count clicks" and creates a "trend chart" the team understands better what Angela will accept for the data to be collected and how she expects it to be analyzed.
  - The user stories would also help Angela when explaining to Dave, her boss, the specific requirements that the team would be focusing on in the next sprint.
- Angela mentioned that expressing the requirements in a user story format demanded more effort from her
  - but after some discussion, she agreed that this small upfront investment was worthwhile because it made her think more about what she wanted.

Applying *Splitting User Stories* Pattern

- As part of preparing user stories for development, the team proceeded to split each user story that was too large
  - The outcome are smaller stories that are more aligned to the INVEST criteria
    - Especially the testable criteria
- Having smaller stories, with clear test criteria, makes each story easier to complete
  - This rewards team members with a sense of achievement, and
  - Improves team member progress assessments.



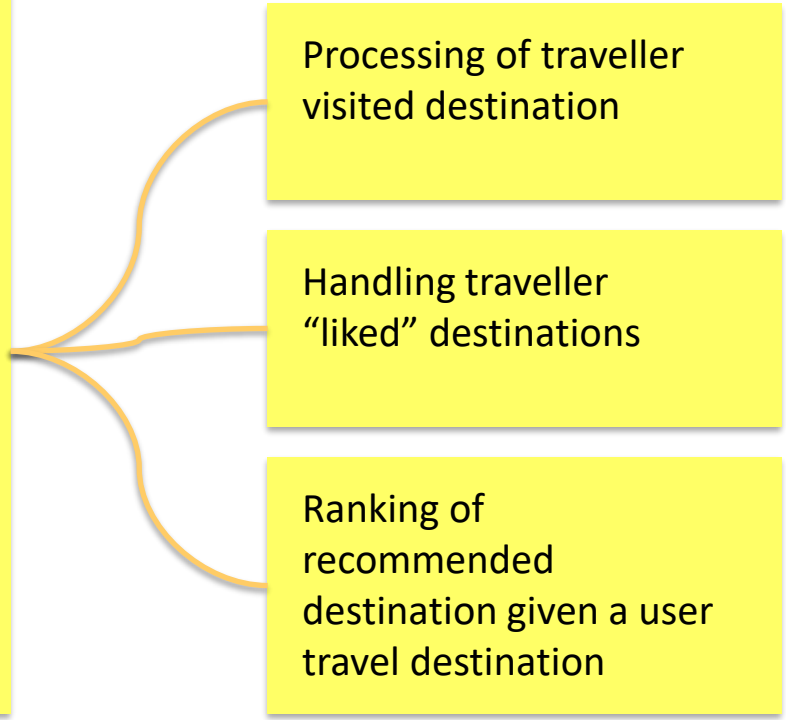
Travel Essence Splitting User Stories

As an example, this Figure shows how the first user story was split into 3 smaller ones

As a traveller, I want to have destinations I like to be ranked higher than other destinations
so that it is easier for me find these destinations

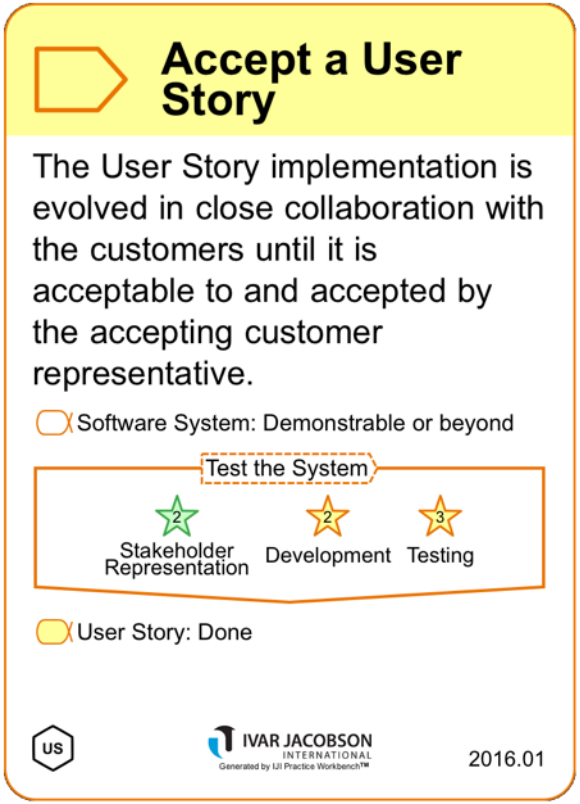
Acceptance criteria:

1. A visited destinations ranks higher than a non-visited one
2. A “liked” destination ranks higher than a “non-liked” destination



Accept a User Story Activity

- The Accept a User Story activity is about having the customer representative accept that the User Story is implemented
  - The card indicates that:
    - The user story alpha achieve the **Done** state

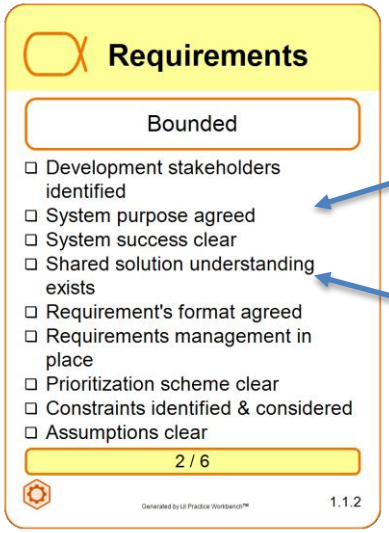


- Acceptance criteria expressed clearly was an investment that paid off as developers had a clearer idea what had to be done
  - The result was reduced disagreements when accepting the story.
- Over the course of the delivery of each user story, they regularly communicated with Angela and with each other regarding the details of the user story.
  - Angela also participated in the acceptance of each user story.
  - Whenever issues arose during the sprint, she worked with the team to refine the acceptance criteria.

The Value of the Kernel to the User Story Practice

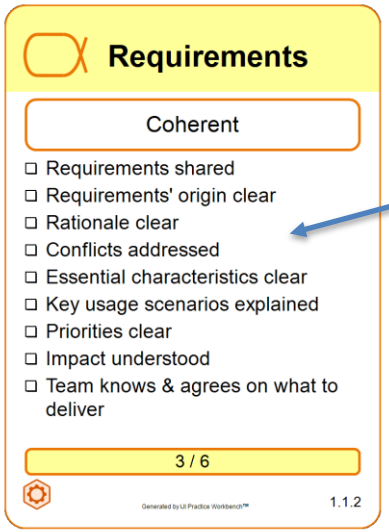
- By describing the User Story practice in an essentialized form (i.e. activity cards showing relationships to alphas being progressed) we can see which alphas is being progressed and where the requirements practice still had weaknesses.
- Specifically the User Story practice helps achieve the following Essence kernel alpha states:
  - **Requirements** alpha: *Bounded* and *Coherent* state
  - **Work** alpha: *Prepared* state
  - **Requirements** alpha: *Acceptable* state

Requirements alpha: Bounded and Coherent state



System purpose agreed

Shared solution understanding exists

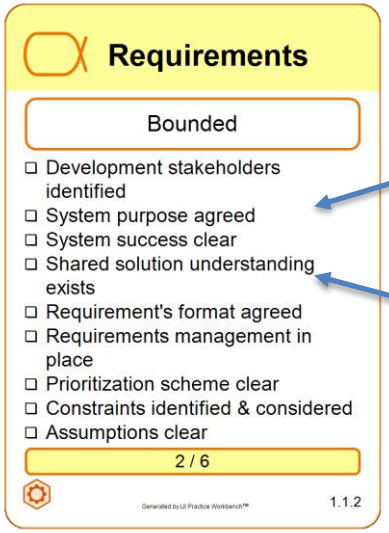


Conflicts addressed

The explicit activities in the user story practice directly supports the team in achieving key checklists within the Requirements alpha: Bounded and Coherent states.

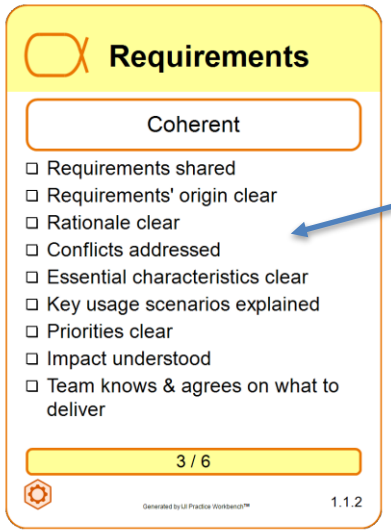
- The User Story practice encouraged stakeholders and team members to discuss and therefore:
  - **agree on the purpose** of the new system,
  - **helping everyone involved to achieve a shared understanding** of the extent of the proposed system.
- Discussions helped both the team members and stakeholders to
  - **work through issues related to potentially conflicting requirements**

Requirements alpha: Bounded and Coherent state



System purpose agreed

Shared solution understanding exists

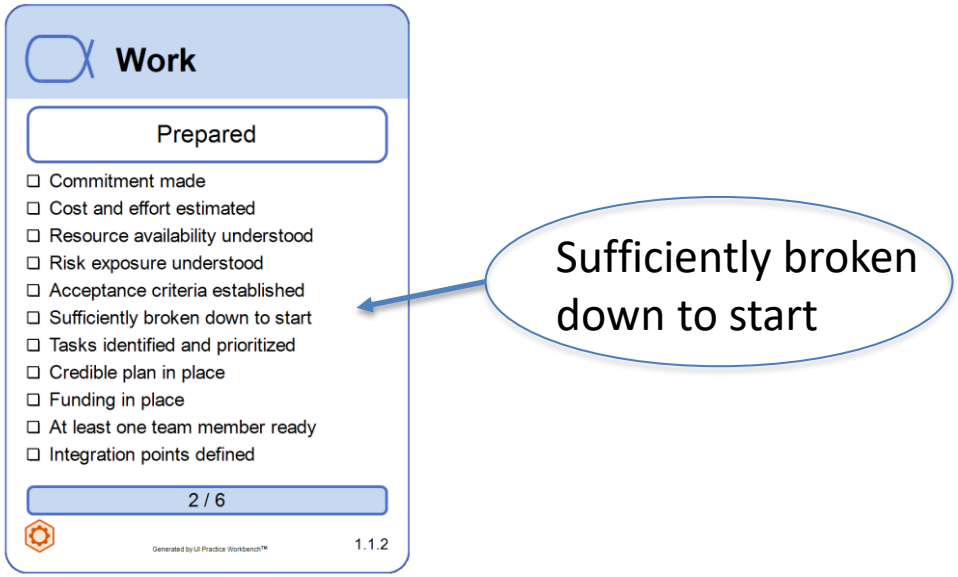


Conflicts addressed

The explicit activities in the user story practice directly supports the team in achieving key checklists within the Requirements alpha: Bounded and Coherent states.

- The User Story practice encouraged stakeholders and team members to discuss and therefore:
  - **agree on the purpose** of the new system,
  - **helping everyone involved to achieve a shared understanding** of the extent of the proposed system.
- Discussions helped both the team members and stakeholders to
  - **work through issues related to potentially conflicting requirements**

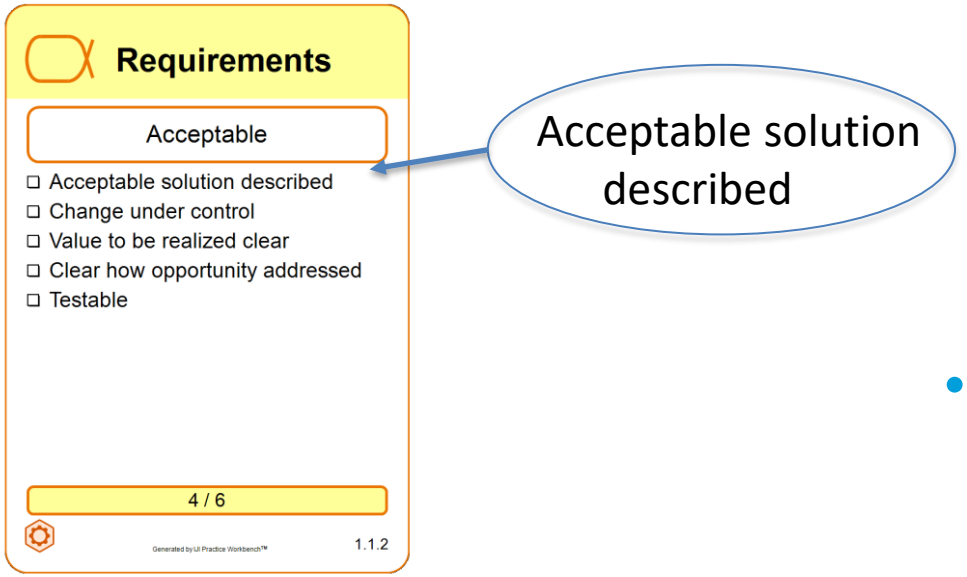
Work alpha: Prepared state



Achieving the Prepared state of the Work alpha was helped by the User Story practice

- Because it encourages the splitting of each user story
  - In order to break the requirements down into tasks that the team could estimate and commit to completing within a single sprint

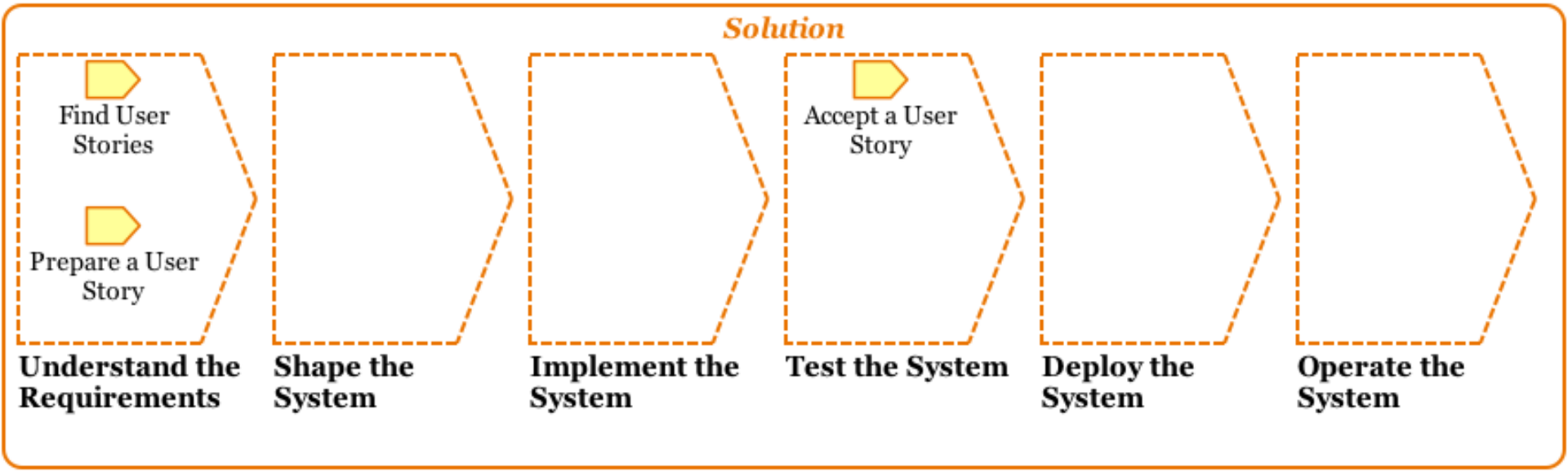
Requirements alpha: Acceptable state



- The explicit activities in the User Story practice directly supported the team in achieving key checklists in the Requirements alpha: Acceptable state.
- The User Story practice encouraged acceptance criteria to be agreed to
  - This lead the team of the importance of describing clear test steps that would lead to an acceptable solution

Impact of User Stories Lite practice for the team

- The three activities in the User Story Lite practice only cover two activity spaces
  - no activity covers the "Shape the System" activity space
    - This is the activity space that deals with the structure of the solution area including the structure of requirements
- By looking at their practices through the lens of Essence, the team can see the strengths of their current practices, as well as the weaknesses



User Stories Lite bridge to Use Case Lite

- The User Stories Lite practice helped the team progress forward with multiple Essence kernel alphas
  - Yet, it did not solve all the challenges the development team faced with regard to progressing the Requirements and the Work alphas to satisfy the Product Owner
  - They had a list of user stories, but not how stories were related to one another.
  - In the next chapter, we will present Use Cases and will discuss pros and cons