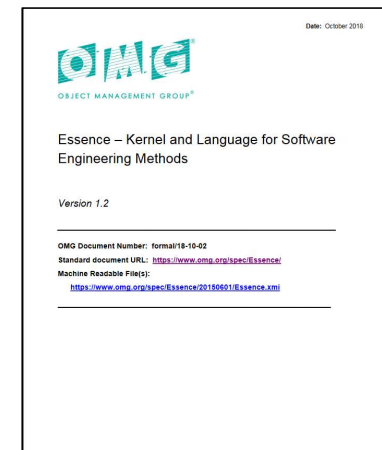
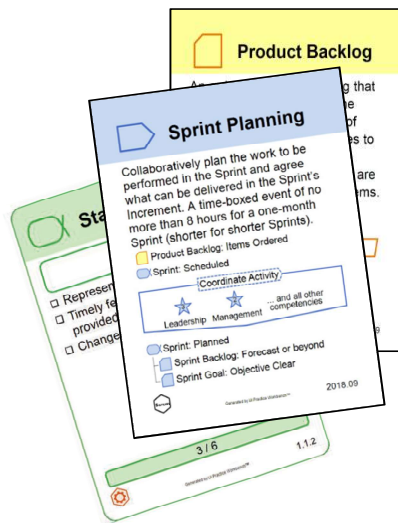




<https://www.scrumguides.org/>

# SCRUM

Essentialized



[OMG2018a] : <https://www.omg.org/spec/Essence>

# From SCRUM to Essence

## Gestion de projet agile avec Scrum

### La formation

Formation en vidéo - introduction à la gestion de projet agile avec Scrum.

Au terme de cette formation, vous devrez être capable :

1. Quand - ce qu'une approche agile ?
2. Fondamentaux concrets de Scrum.
3. Les 8 leviers de réussite agiles.
4. Le rôle de Scrum Master.
5. Animation de la rétrospective.

Sur la page [impasse des projets](#), vous pouvez télécharger les différents des fichiers pptx et pdf de la formation, ainsi que les fichiers vidéos. Enfin, voici un [document de police de notes de cours](#) portable.

### Objectifs du cours



Module DP45 - Introduction

### 3 piliers de Scrum



Le quiz n'est actuellement disponible que sur le MOOC GdP.

Notes : 3 piliers

- ▶ 3 processus empiriques fondamentaux

### Processus Scrum



Le quiz n'est actuellement disponible que sur le MOOC GdP.

Notes : Le processus SCRUM

- ▶ Fonctionnement concret de Scrum

### Démarche itérative et incrémentale



Le quiz n'est actuellement disponible que sur le MOOC GdP.

Notes : Synthèse du chapitre

- ▶ Synthèse

### Le rôle de Scrum Master 4/4



Le quiz n'est actuellement disponible que sur le MOOC GdP.

Notes : Synthèse

- ▶ Synthèse

### Animation de la rétrospective



Le quiz n'est actuellement disponible que sur le MOOC GdP.

Notes : La rétrospective

- ▶ Déroulement
- ▶ Commencer par les choses positives
- ▶ Ce qui n'a pas bien marché
- ▶ Plan d'action
- ▶ Autre méthode visuelle

### Conclusion



Le quiz n'est actuellement disponible que sur le MOOC GdP.

Notes : Conclusion

- ▶ Caractéristiques de SCRUM

### Qu'est-ce qu'une approche agile - 1?



Le quiz n'est actuellement disponible que sur le MOOC GdP.

Notes : Qu'est-ce qu'une approche agile ?

- ▶ Le manifeste
- ▶ Les 12 principes
- ▶ Le constat
- ▶ SCRUM: diviser pour mieux maîtriser

### Qu'est-ce qu'une approche agile - 2?



Le quiz n'est actuellement disponible que sur le MOOC GdP.

Notes : Qu'est-ce qu'une approche agile ?

- ▶ Synthèse

### Les 8 leviers de réussite 1/3



Le quiz n'est actuellement disponible que sur le MOOC GdP.

### Le rôle de Scrum Master 1/4



Le quiz n'est actuellement disponible que sur le MOOC GdP.

Notes : Les acteurs

### Les 8 leviers de réussite 2/3



Le quiz n'est actuellement disponible que sur le MOOC GdP.

### Le rôle de Scrum Master 2/4



Le quiz n'est actuellement disponible que sur le MOOC GdP.

Notes : Chef de projet Scrum vs chef de projet traditionnel

### Les 8 leviers de réussite 3/3



Le quiz n'est actuellement disponible que sur le MOOC GdP.

### Le rôle de Scrum Master 3/4



Le quiz n'est actuellement disponible que sur le MOOC GdP.

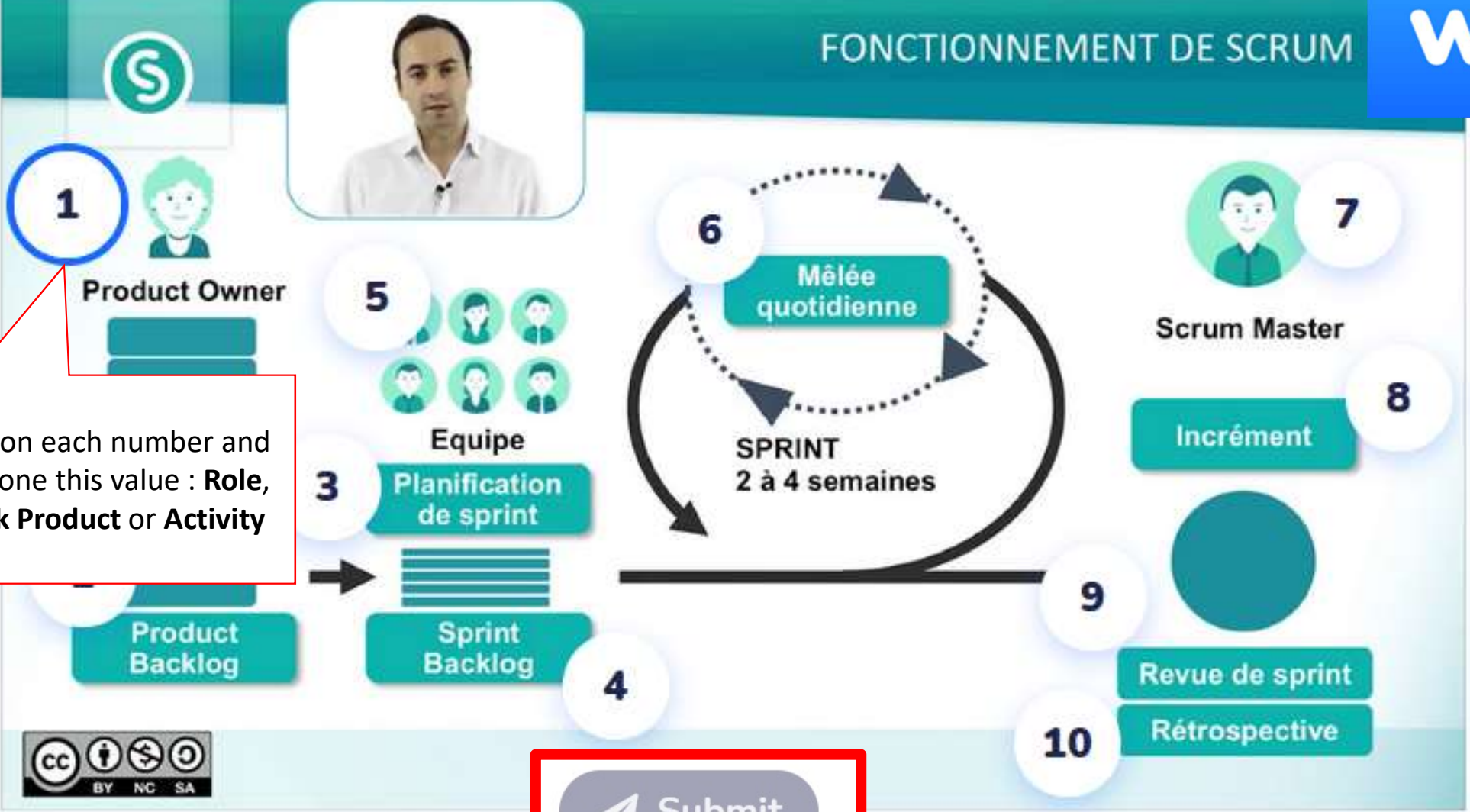
Notes : Scrum Master

Essence standard gives the opportunity to reformulate what has been learned with the SCRUM e-learning.

<https://app.wooclap.com/RGTKHF>



# FONCTIONNEMENT DE SCRUM

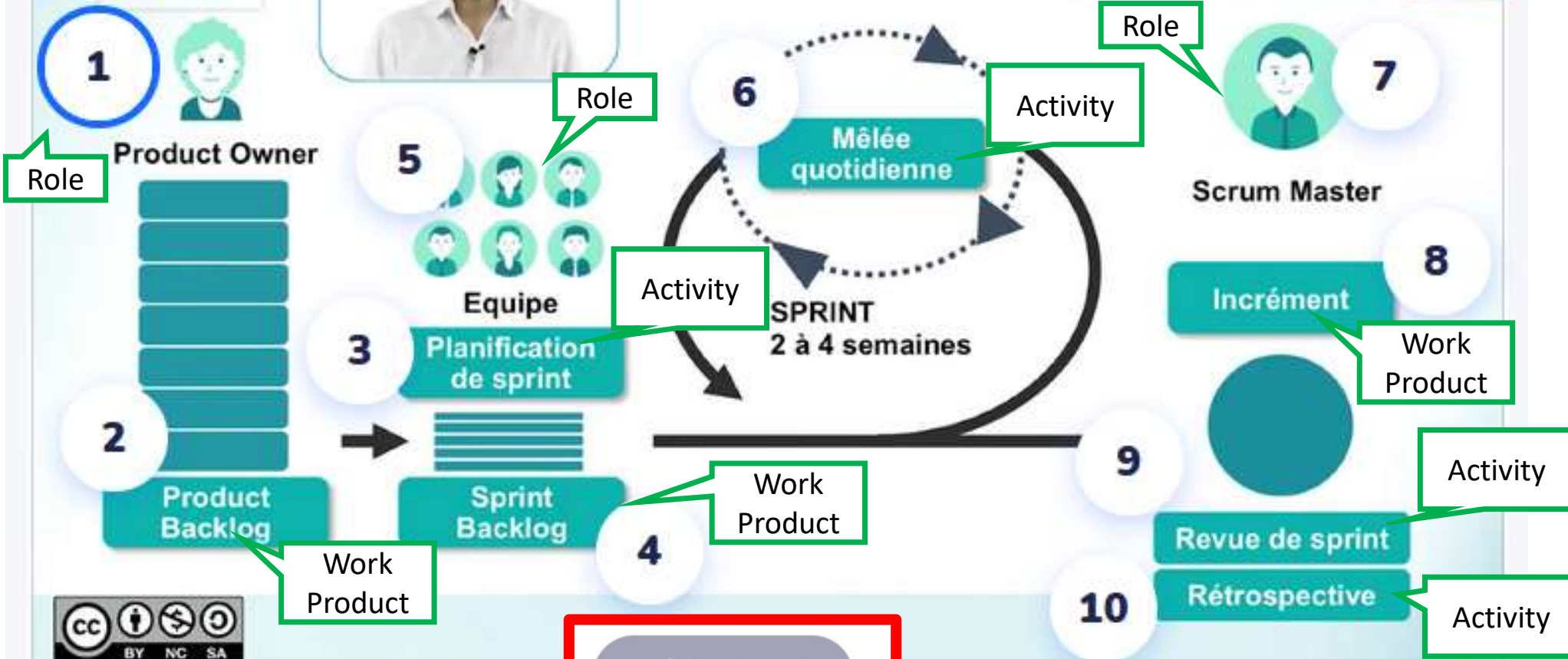


Click on each number and type one this value : **Role**, **Work Product** or **Activity**

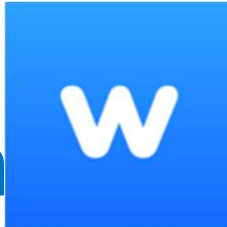


Submit

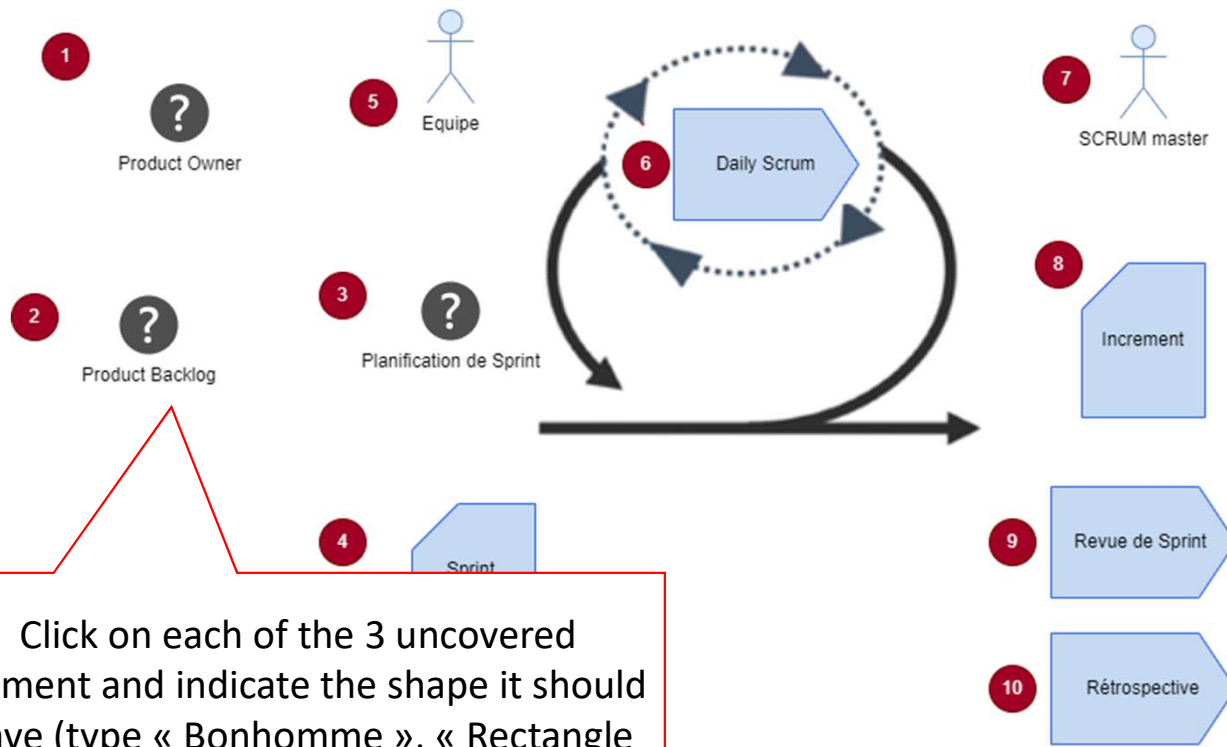
# FONCTIONNEMENT DE SCRUM



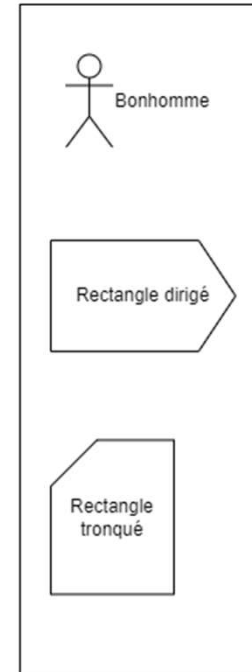
# Essence graphical notation



# SCRUM : From unformal to Essence notation

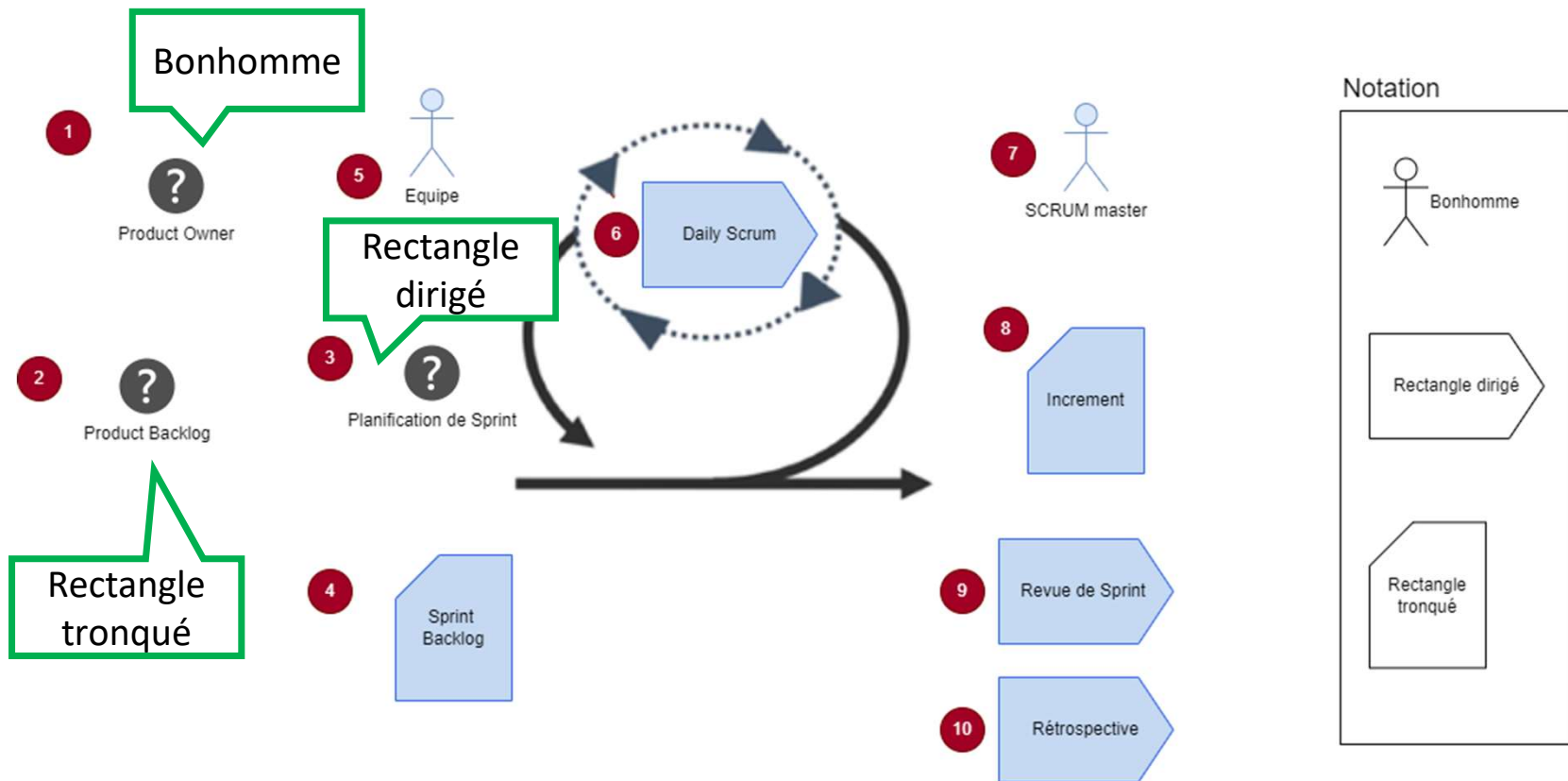


Notation



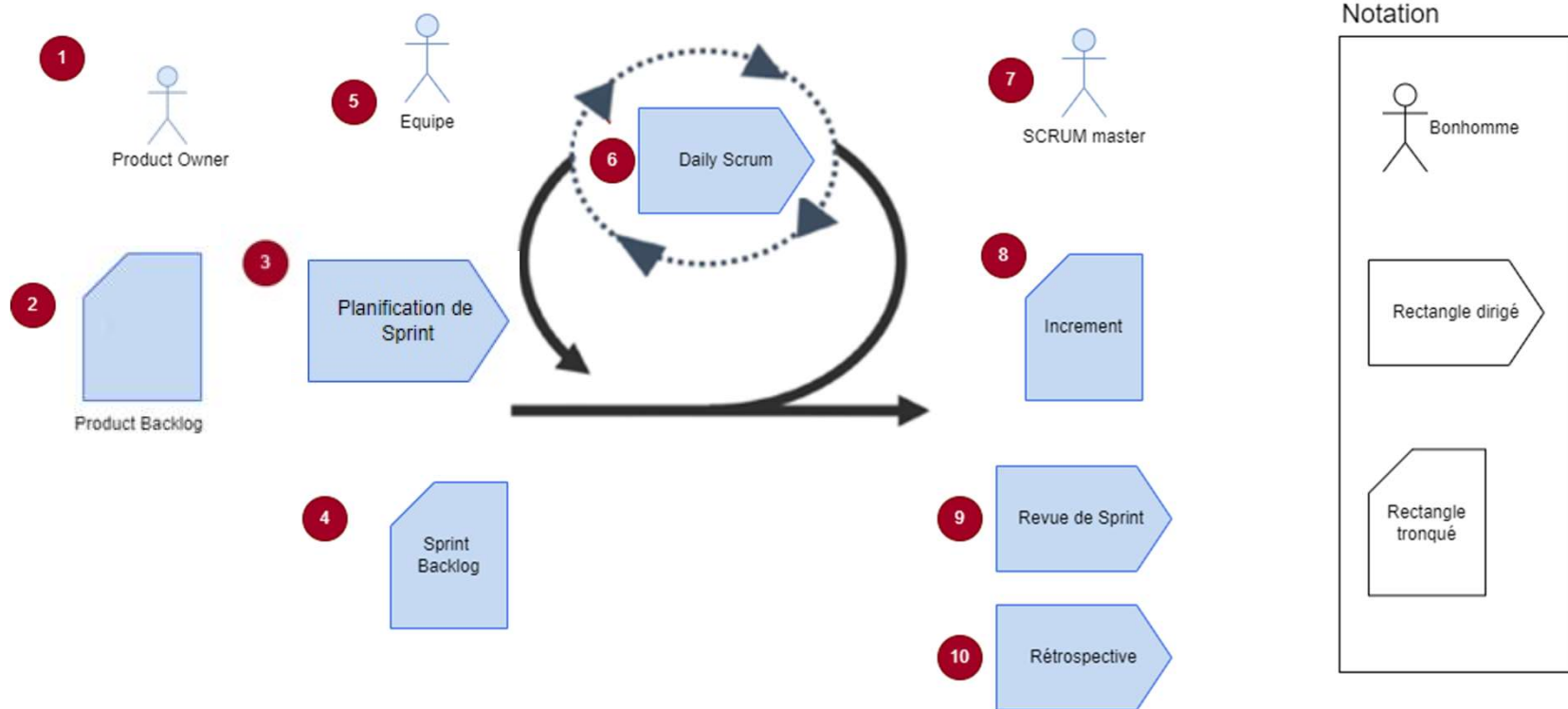
Click on each of the 3 uncovered element and indicate the shape it should have (type « Bonhomme », « Rectangle dirigé », or « Rectangle tronqué »)

# SCRUM : From unformal to Essence notation





# SCRUM : From unformal to Essence notation



# Essence cards

# Cards

fr : Planification de Sprint

3 cards (out of a 10 cards set)

## Sprint Planning



Collaboratively plan the work to be performed in the Sprint and agree what can be delivered in the Sprint's Increment. A time-boxed event of no more than 8 hours for a one-month Sprint (shorter for shorter Sprints).

- Product Backlog: Items Ordered
- Sprint: Scheduled

**Coordinate Activity**

Leadership Management ... and all other competencies


- Sprint: Planned
- Sprint Backlog: Forecast or beyond
- Sprint Goal: Objective Clear



 Generated by  2018.09

## Product Owner

The Product Owner is responsible for maximizing the value of the product resulting from the work of the Development Team. They are the sole person responsible for managing the Product Backlog ensuring:

- The Product Backlog Items are clearly expressed
- The Product Backlog is ordered, visible, transparent and clear to all
- The development team understand the Product Backlog Items.


Part of:  Scrum Team



 Generated by  2018.09

## Product Backlog

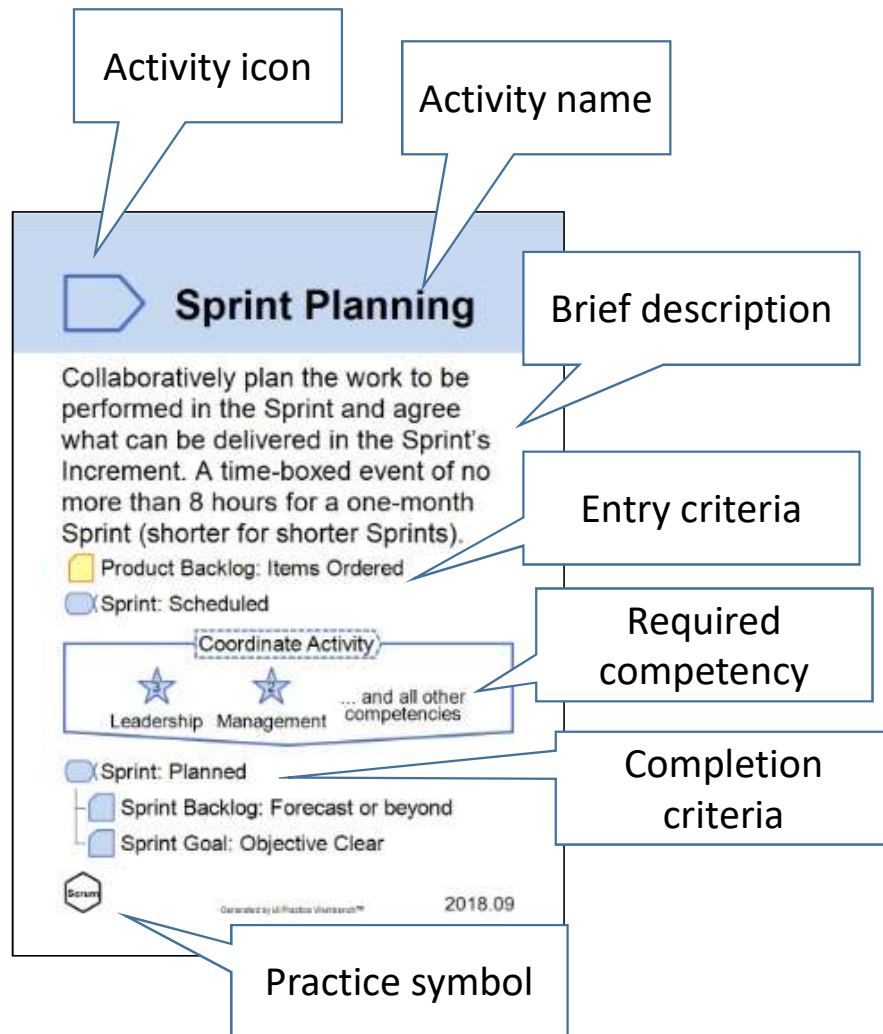
An ordered list of everything that is known to be needed in the product. The single source of requirements for any changes to be made to the product. The items in the Product Backlog are known as Product Backlog Items.

Items Ordered

Describes:  Requirements

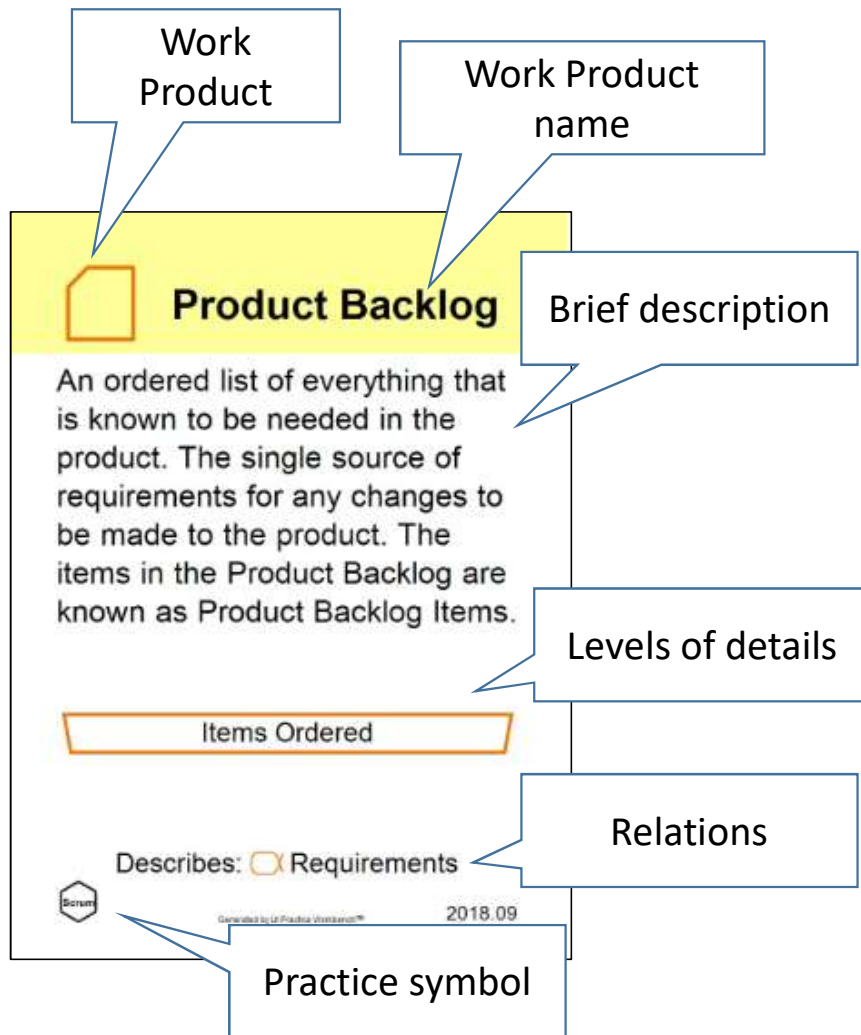
 Generated by  2018.09

# Activity card



- The card have a specific layout for each type (here is card layout for Activity)

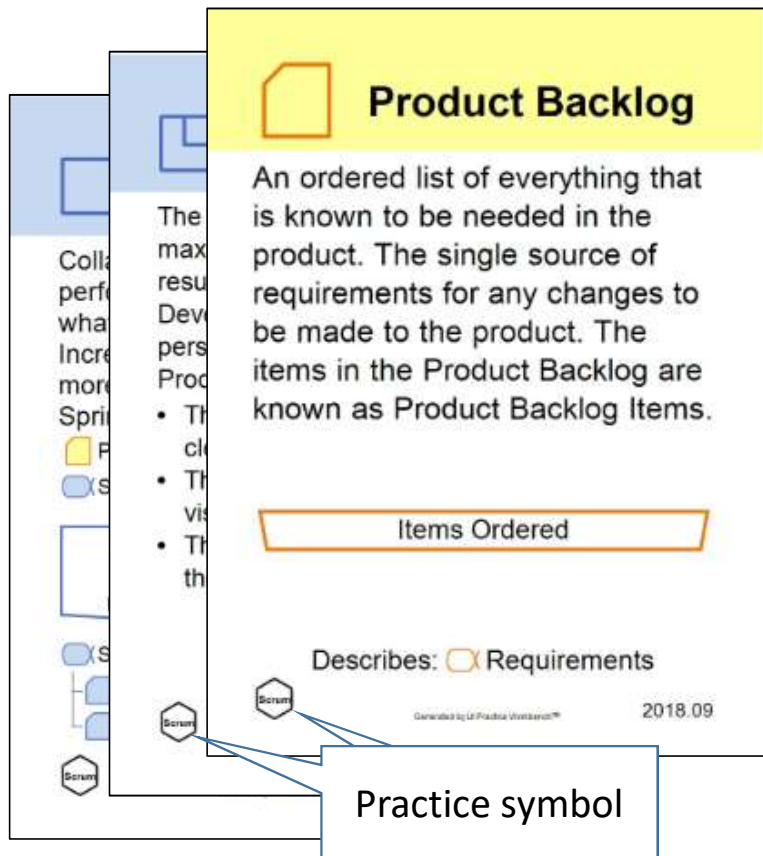
# Work Product card



# Essence practices and coverage

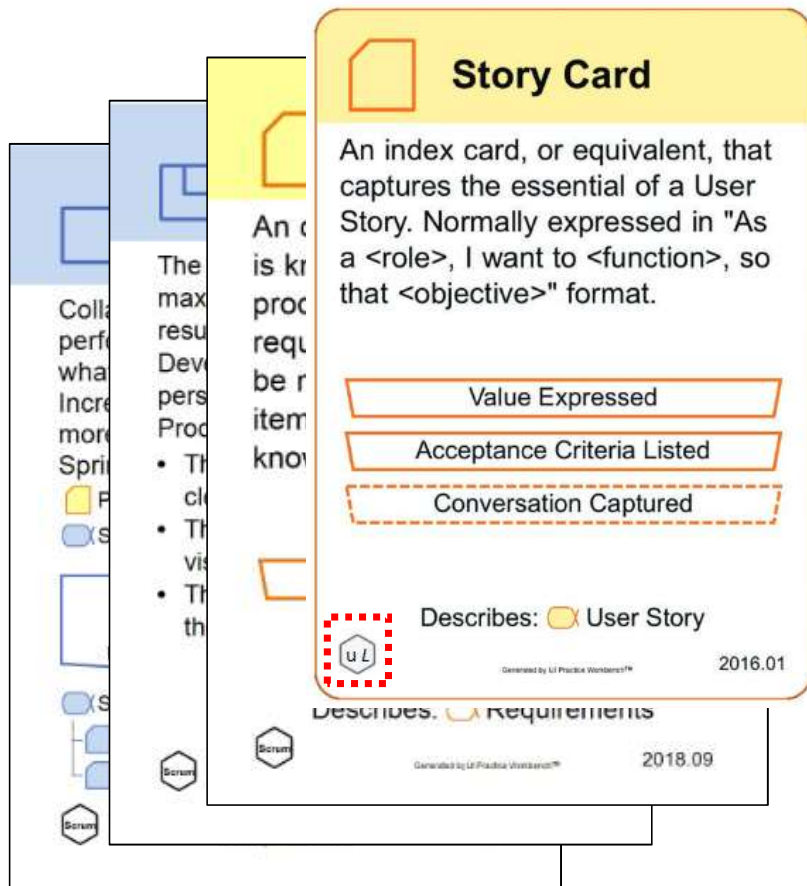
SCRUM, User Story Lite, ...

# Essence as a cards and practices ecosystem (1)



-  SCRUM practice

# Essence as a cards and practices ecosystem (2)

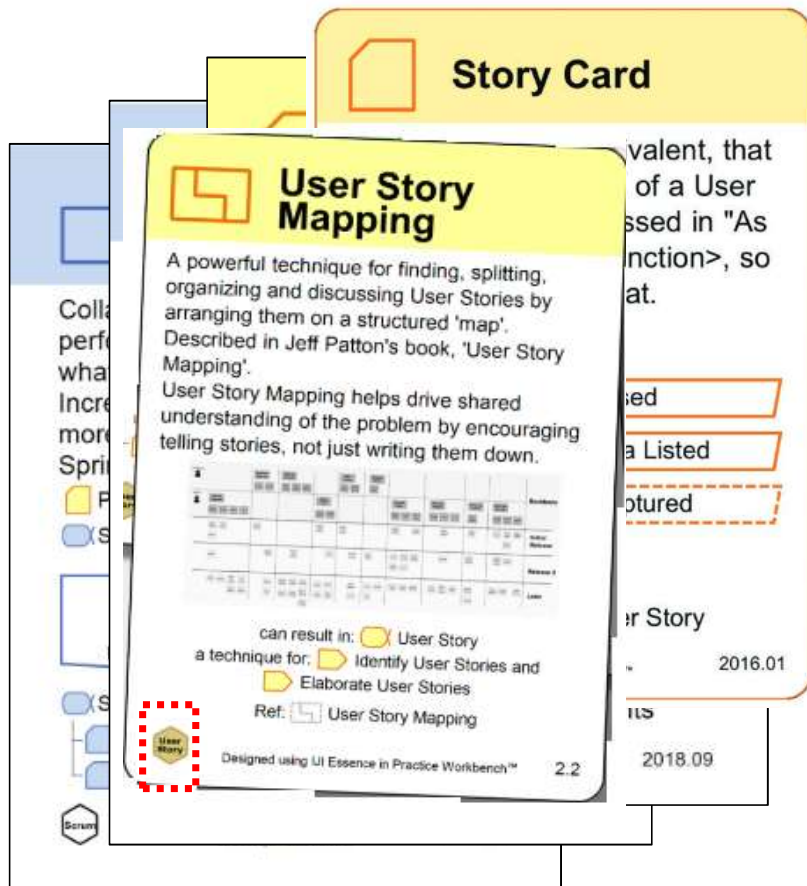


-  SCRUM practice

-  User's Story Lite



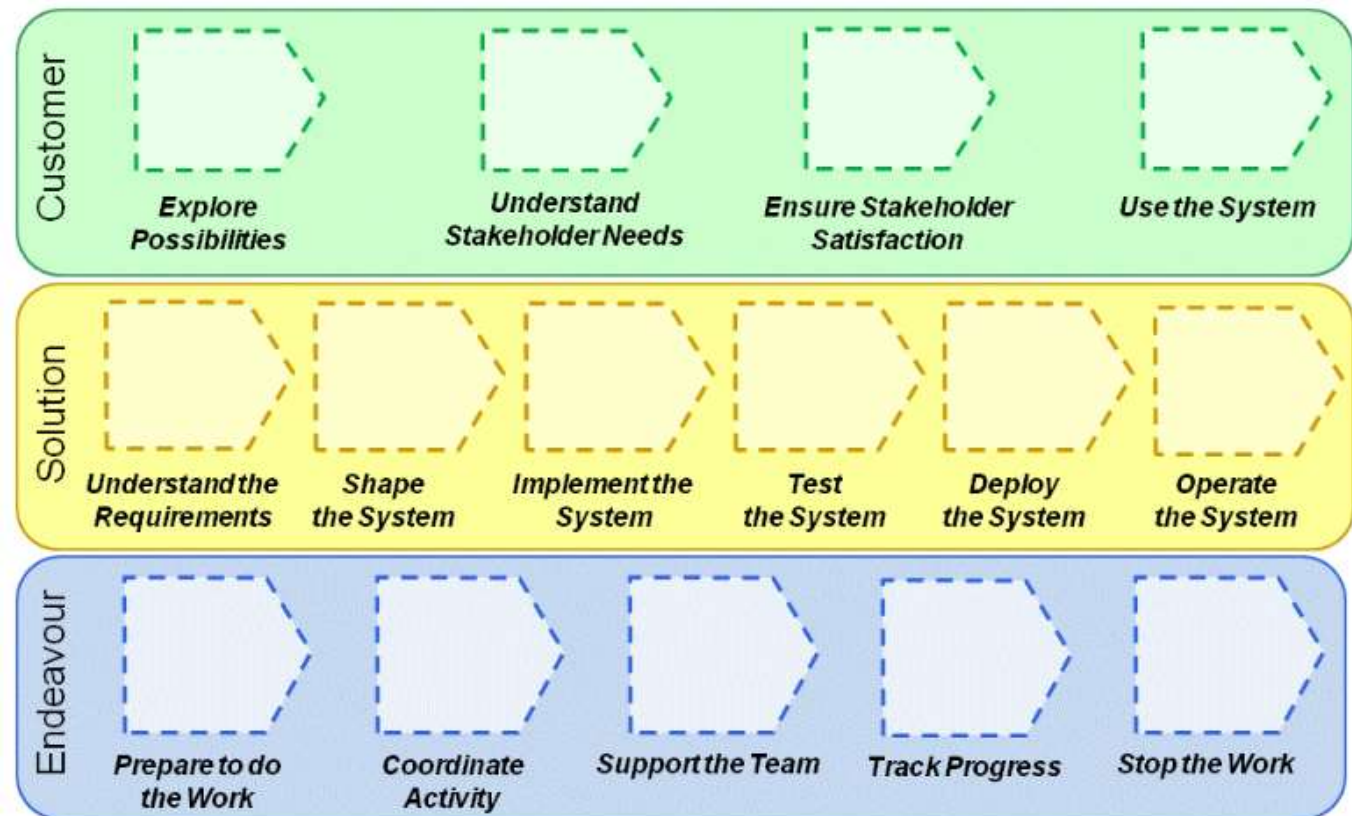
# Essence as a cards and practices ecosystem (3)



- SCRUM practice
- User's Story Lite
- User's Story Advanced
- Use Case Lite
- Spotify practice

# Activities coverage of practices

- Essence proposes an activity based view of software engineering
- All the activities of a software project are divided into **Activities Spaces**.
- It can help to see what is covered by a given practice



Source : [OMG2018a] p18, Figure 8.3 – The Kernel Activity Spaces

# Coverage : SCRUM practice

**Sprint Retrospective**

The whole team meets regularly to reflect on its way of working. Improvements are identified and prioritized, and actions agreed. At the next retrospective the results are evaluated.

Support the Team

Leadership Management

**Sprint Planning**

Decide what can be delivered in the Sprint's Increment and how the work needed to deliver the agreement will be achieved.

Sprint: Scheduled

**Daily Scrum**

The team meets every day, same time and place, to assess progress, synchronize activity, and raise and action impediments. The meeting is time-boxed, typically to 15 minutes.

Gather all the Activity cards of the practice under study

**Sprint Review**

A time-boxed review of the outcomes of the Sprint to gather feedback and discuss what should be done next.

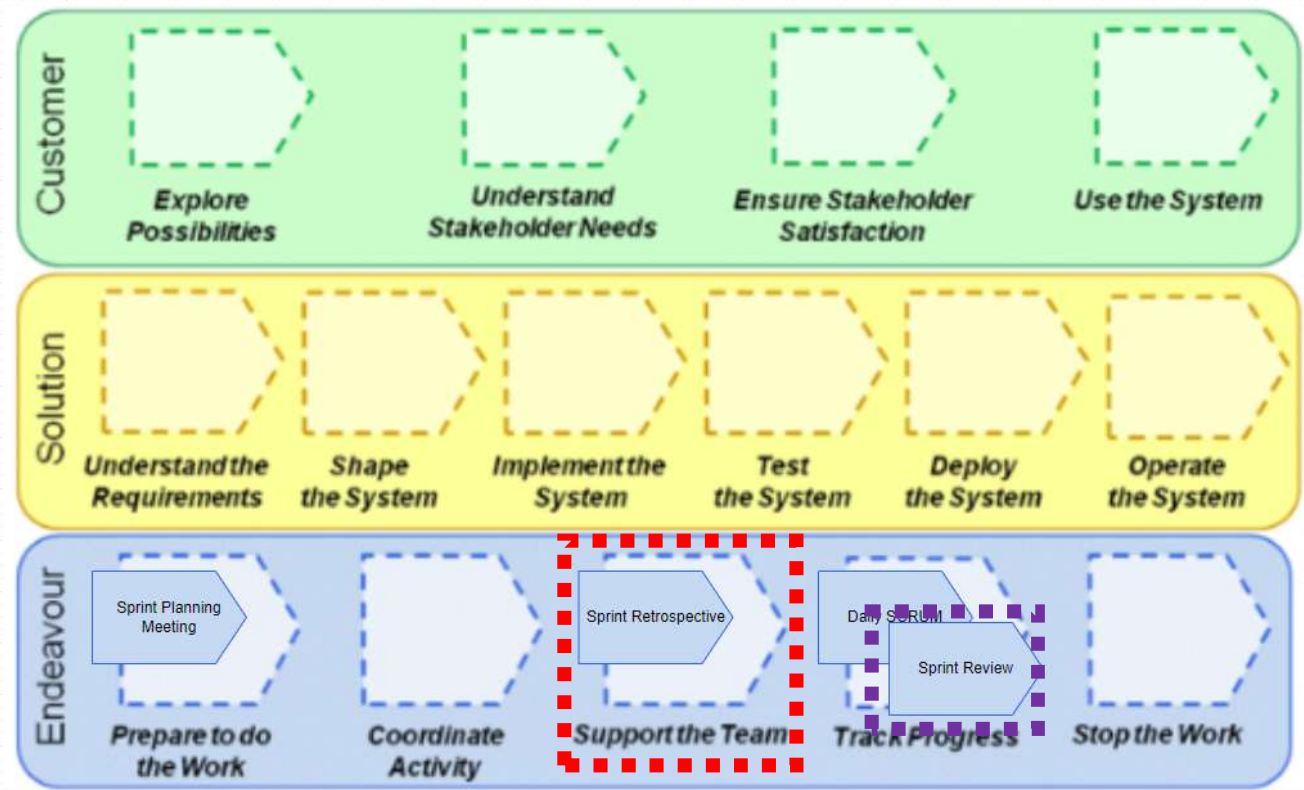
- Product Backlog: at any level
- Sprint: Planned

Track Progress

Development Management

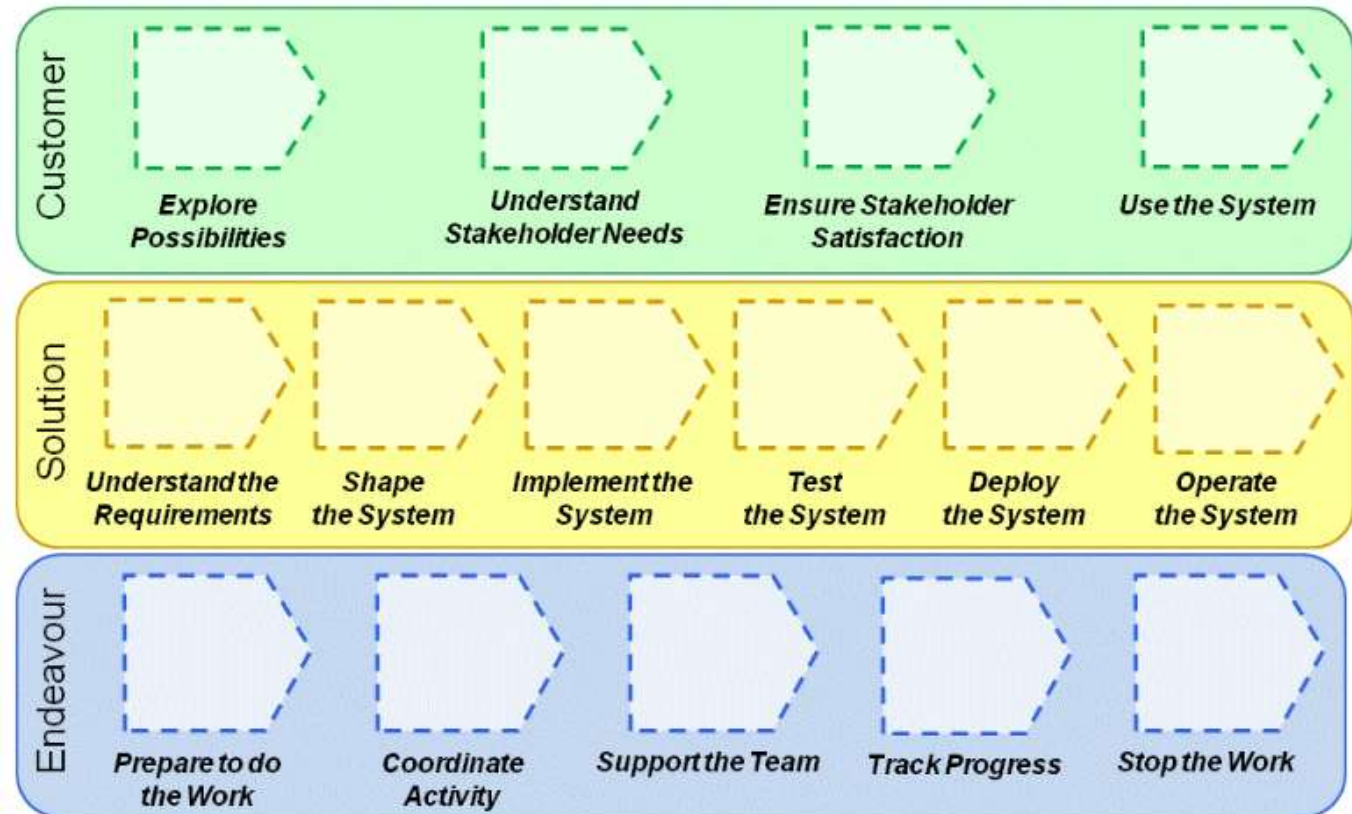
- Increment: Product Backlog Items Listed
- Sprint: Reviewed

03.2015



# Coverage : How to identify missing practices ?

- Some practices may require **additional** practices to cover all the needs of a project team.



Source : [OMG2018a] p18, Figure 8.3 – The Kernel Activity Spaces

# Coverage : How to identify missing practices ?

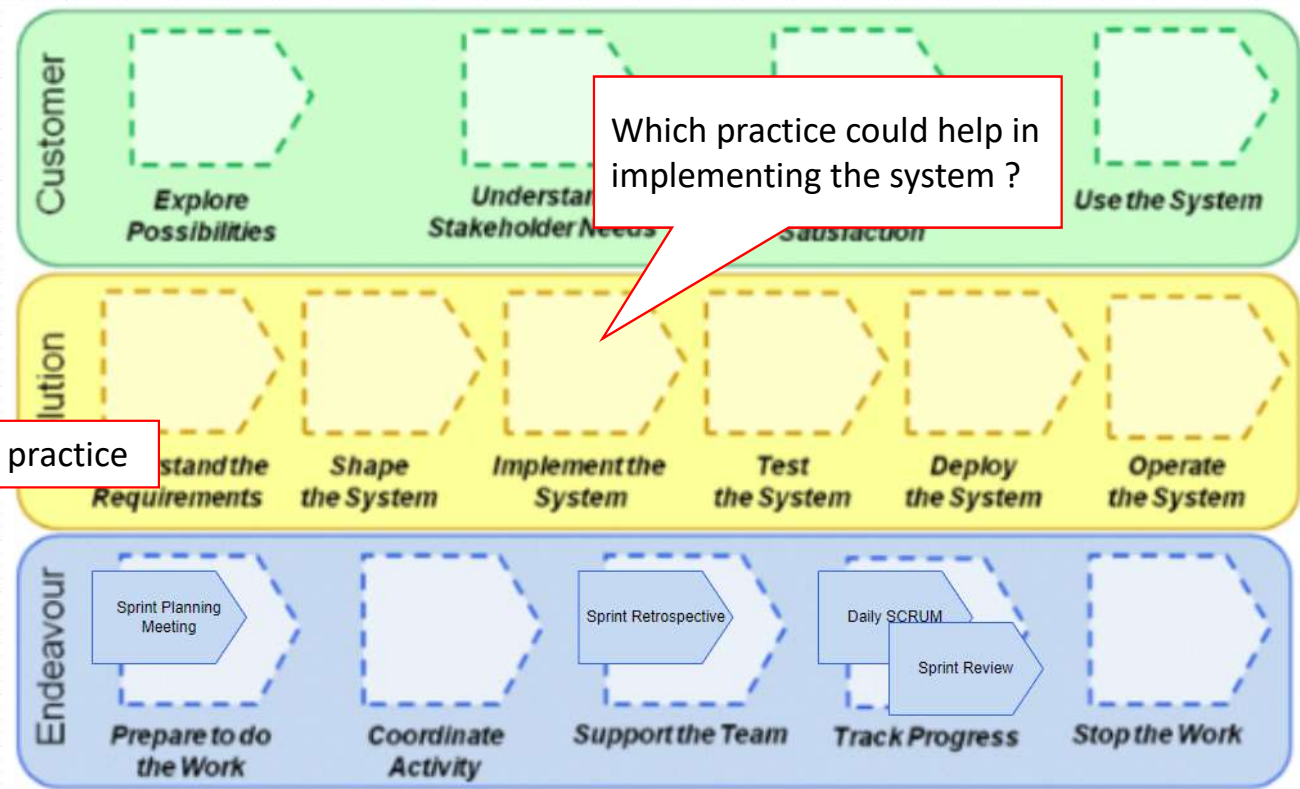
**Find Actors Use Cases**  
 Agree on the goals and the system by identifying using and testing it.

**Write Code**  
 Collaborate together to produce good quality code that meet requirements.

**Test a Use-Slice**  
 Verify the slice is done for inclusion in a release.

**Use Case Lite practice** (CL icon)

**Programming Lite practice** (P icon)



Which practice could help in implementing the system ?



# Coverage : Implement the system

Which practice could help in implementing the system ?

**Find Actors and Use Cases**

Agree on the goals and value of the system by identifying ways of using and testing it.

Understand the Requirements

Stakeholder Representation Analysis

- Requirements: Conceived (contributes to) - Bounded (contributes to) - Coherent (contributes to)
- Use-Case Model: Value Established or beyond
- Use Case: Goal Established
- Use-Case Narrative: Briefly Described or beyond

CL 5.2.0

**Slice the Use-Cases**

Break use case up into a number of intelligently selected smaller parts for development.

Shape the System

Development Analysis Testing

- Requirements: Coherent
- Use-Case Model: Structured
- Use Case: Story Structure Understood
- Use-Case Slice: Scoped

CL 5.2.0

**Prepare a Use-Case Slice**

Enhance the narrative and test cases to clearly define what it means to successfully implement the slice.

Understand the Requirements

Analysis Testing Development

- Use Case: Story Structure Understood
- Test Case: Scenario Chosen
- Use-Case Narrative: Essential Outline
- Use-Case Slice: Prepared

CL 5.2.0

**Test a Use-Case Slice**

Verify the slice is done and ready for inclusion in a release.

Test the System

Analysis Testing

- Requirements: Addressed or beyond
- Use Case: Simplest Story Fulfilled or beyond
- Test Case: Variables Set
- Use-Case Slice: Verified

CL 5.2.0

**Write Code**

Collaborate together to produce good quality code that meet requirements.

Requirements: Bounded

Implement the System

Development

- Requirements: Addressed
- Software System: Ready
- Code: Code Completed

P 5.2.0



# Coverage : Implement the system

**Find Actors and Use Cases**

Agree on the goals and value of the system by identifying ways of using and testing it.

Understand the Requirements

- Stakeholder Representation
- Analysis

- Requirements: Conceived (contributes to) - Bounded (contributes to) - Coherent (contributes to)
- Use-Case Model: Value Established or beyond
- Use Case: Goal Established
- Use-Case Narrative: Briefly Described or beyond

CL 5.2.0

**Slice the Use-Cases**

Break use case up into a number of intelligently selected smaller parts for development.

Shape the System

- Development
- Analysis
- Testing

- Requirements: Coherent
- Use-Case Model: Structured
- Use Case: Story Structure Understood
- Use-Case Slice: Scoped

CL 5.2.0

**Prepare a Use-Case Slice**

Enhance the narrative and test cases to clearly define what it means to successfully implement the slice.

Understand the Requirements

- Analysis
- Testing
- Development

- Use Case: Story Structure Understood
- Test Case: Scenario Chosen
- Use-Case Narrative: Essential Outline
- Use-Case Slice: Prepared

CL 5.2.0

**Test a Use-Case Slice**

Verify the slice is done for inclusion in a release.

Test the System

- Analysis
- Testing

- Requirements: Addressed
- Use Case: Simplest Scenario beyond
- Test Case: Variables Set
- Use-Case Slice: Verified

CL 5.2.0

**Write Code**

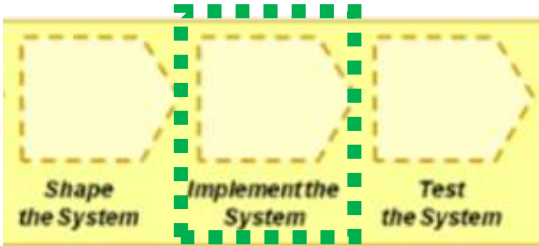
Collaborate together to produce good quality code that meet requirements.

Implement the System

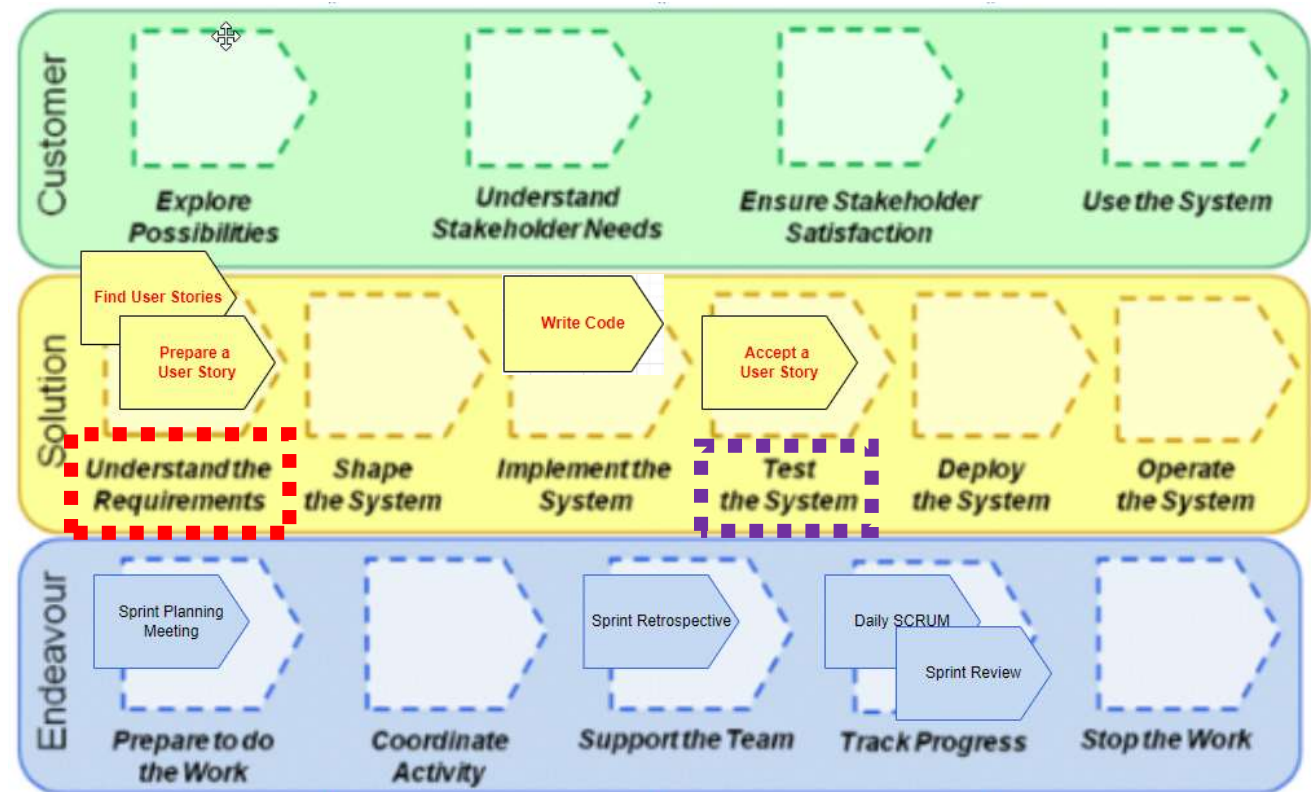
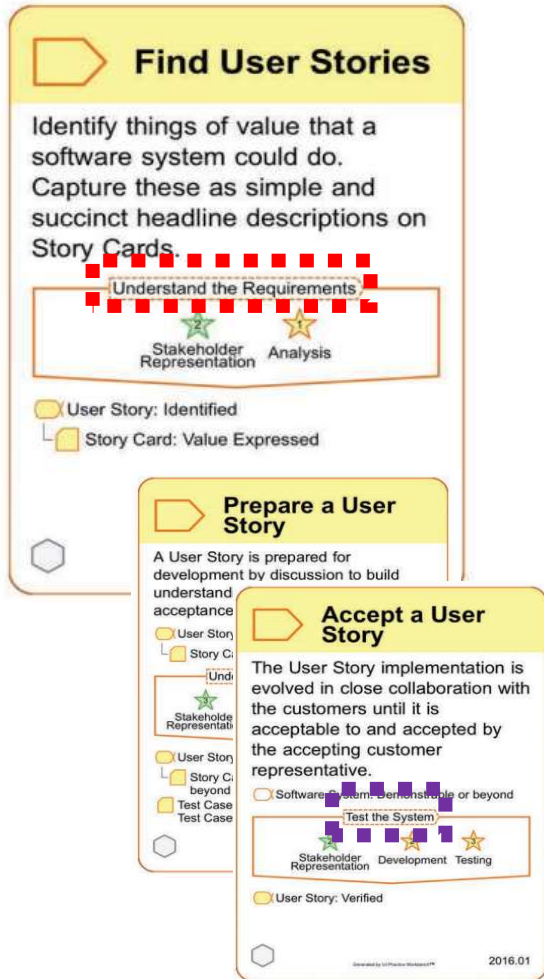
- Requirements: Bounded
- Development

- Requirements: **P**ressed
- Software System: Ready
- Code: Code Completed

CL 5.2.0



# Coverage : User Story Practice

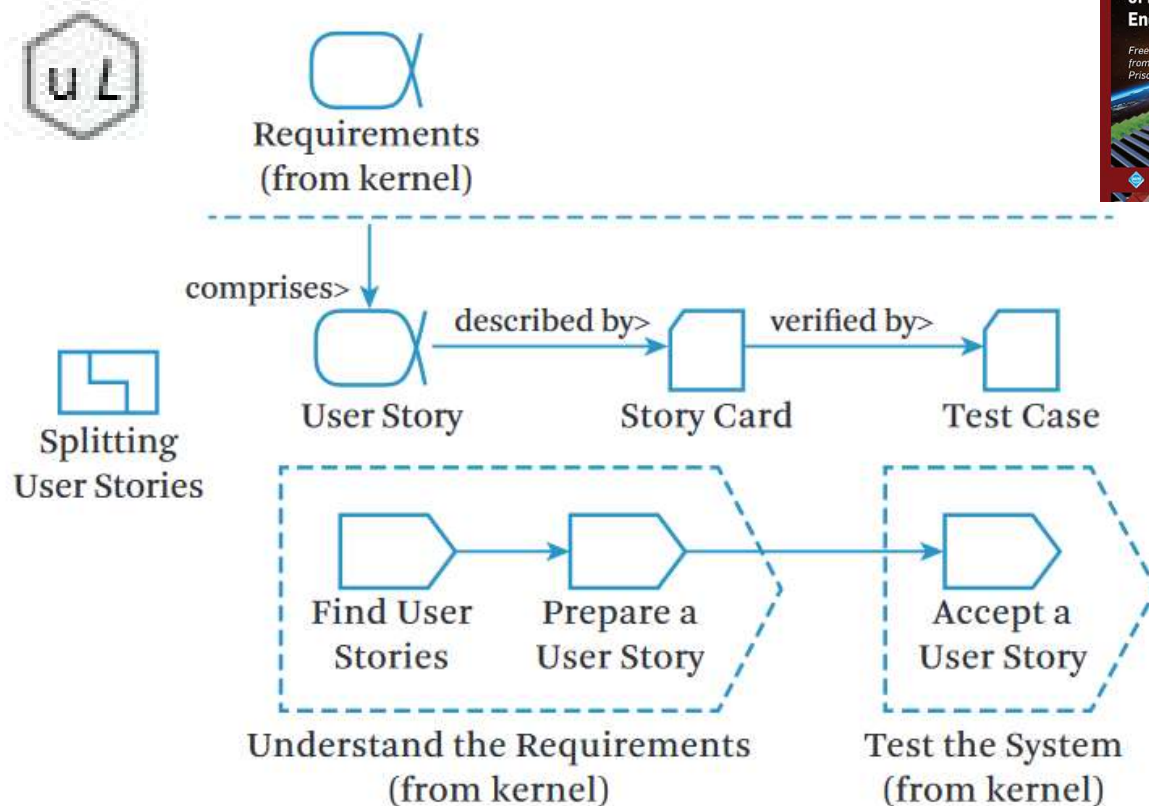
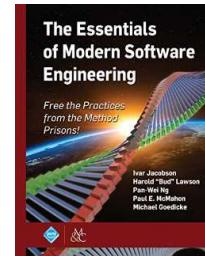




Focus on User Story Lite practice

# User Story Lite : the big picture

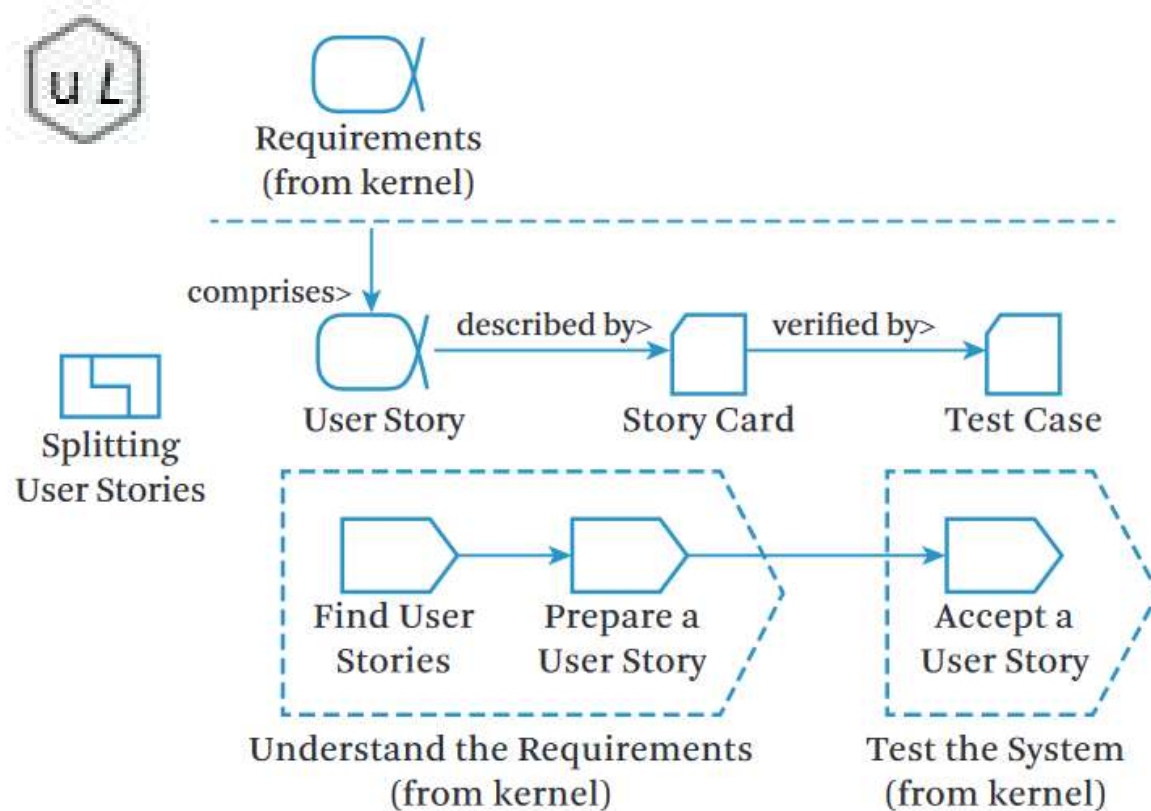
Extrait de [McMahon2019]



- Elements of « User Story Lite » practice, in complement of activities.



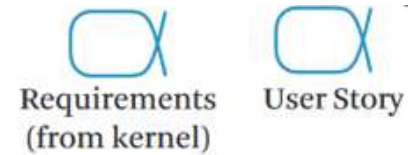
# User Story Lite : the big picture



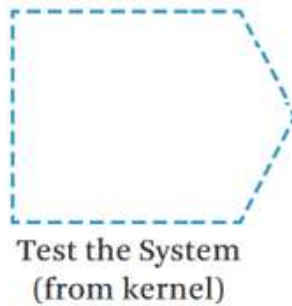
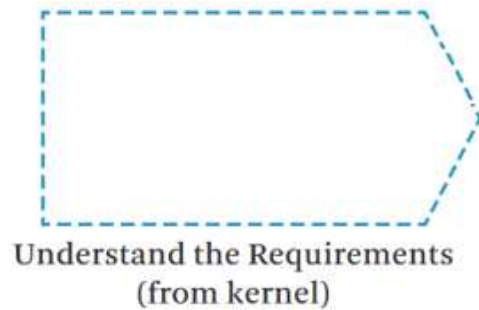
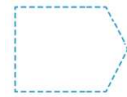
- Where are the new icons ?



# User Story Lite : the big picture



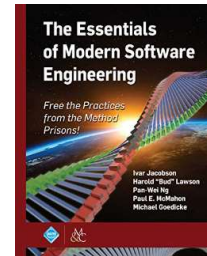
## New icons



- Alpha
- Pattern
- Activity Space

# User Story Lite : focus on Story Card

Extrait de [McMahon2019]



Requirements  
(from kernel)

comprises>

described by>

verified by>

User Story

Story Card

Use Case

Splitting  
User Stories

Find User  
Stories

Prepare a  
User Story

**Story Card**

An index card, or equivalent, that captures the essential of a User Story. Normally expressed in "As a <role>, I want to <function>, so that <objective>" format.

Value Expressed

Acceptance Criteria Listed

Conversation Captured

Describes: User Story


2016.01

Understand the Requirements  
(from kernel)

em

- Elements of « User Story Lite » practice, in complement of activities.

# User Story Lite : Story Card (Work Product)


 **Story Card**


An index card, or equivalent, that captures the essential of a User Story. Normally expressed in "As a <role>, I want to <function>, so that <objective>" format.

Value Expressed

Acceptance Criteria Listed

Conversation Captured

Describes:  User Story

 Copyright © The Agile Alliance 2016.01



Name: ID: Owner:


As a:

I want to:

So I can:

Priority: Size:

# User Story Lite : focus on Story Card


 **Story Card**


An index card, or equivalent, that captures the essential of a User Story. Normally expressed in "As a <role>, I want to <function>, so that <objective>" format.

Value Expressed

Acceptance Criteria Listed

Conversation Captured

Describes:  User Story

 Copyright © Tobias Fertig 2016 2016.01



Name: ID: Owner:


As a:

I want to:

So I can:

Priority: Size:

# User Story Lite : EXG3275




## Story Card


An index card, or equivalent, that captures the essential of a User Story. Normally expressed in "As a <role>, I want to <function>, so that <objective>" format.

Value Expressed

Acceptance Criteria Listed

Conversation Captured



Describes:  User Story

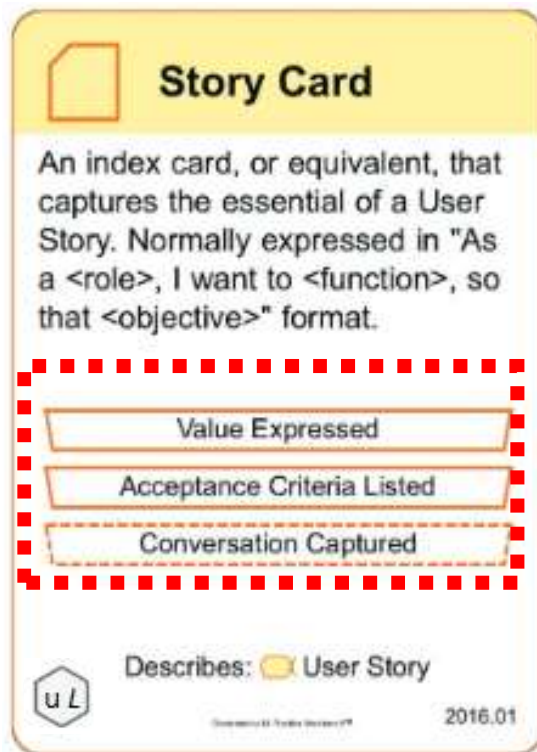
Copyright © 2016 Scrum.org
2016.01

Enoncé : Tableau « Suscheck-Fuqua » - 9 bonnes pratiques concernant les récits utilisateurs (ou user stories)

BF ID	Texte de la bonne pratique
A	<p><b>Excessive 'So That' (Misplaced Requirement)</b>  <b>Problem:</b> The team typically expects details of a story to be found before the conjunction "so that," which is used to explain the story's value. When the product owner writes requirements in the "so that" section, it is easy to miss part of the real requirement since it's hidden after "so that." You can tell if there is a problem when the "so that" section is complex or has multiple parts. In this case, the true story may be so big that it can't get done in a sprint.  <b>Improvement:</b> Move the misplaced needs (from « so that » to « I want ») and analyze a bit more to find a real business value to be written after « so that ».</p>
B	<p><b>Odyssy (Ultra-Huge Story)</b>  <b>Problem:</b> If team members can't even estimate a story at a gross level, they may take on too much work in the iteration and not get to done. An Odyssy is beyond an epic. It's something compounded or diffused to the point of having no discernable value (which an epic has). Such a story will lead to long conversations with the product owner, or even a failure to get to done. The team's velocity will be unpredictable, and team members might be frustrated with a constantly evolving story. The story is too very large to be estimated and hard to implement in a sprint. It's difficult to see the value that holds this story (or epic) together.  <b>Improvement:</b> Stories containing "and" or "or" are likely candidates for splitting into several smaller stories.</p>
C	<p><b>Waterfall</b>  <b>Problem:</b> Stories that contain only analysis, design, or technical aspects lead to waterfall development in two-week phases. The story has no business value and a user who is not really a system end user. There is only the technical side of the equation. This story will likely lead to coding, and no testing since it has no business value.  <b>Improvement:</b> Remove this developer story and split it in many tasks</p>
D	<p><b>Rigidity (Inflexible)</b>  <b>Problem:</b> Stories with too much detail are often inflexible, leaving little room for creativity, better solutions, or dynamic scope control during development. Avoid this by postponing decisions on details that constrain the solution or that specify an implementation. Defer these decisions to a later but still responsible moment in order to maintain maximum flexibility. The product owner is likely to get exactly what he asks for instead of what he really needs.  <b>Improvement:</b> There may be better solutions that leave more flexibility. Specifying this level of technical detail handcuffs the programmers and possibly limits innovation.</p>
E	<p><b>For Whom? (Non-User)</b>  <b>Problem:</b> There are a lot of different types of non-user stories, like using a specific name, a role, or the system.  <b>Improvement:</b> It's better to write user stories for the role that actually wants the benefit or value provided by the user story.</p>
F	<p><b>For Whom? (Generic User)</b>  <b>Problem:</b> the most overused and overly vague person: "As a user" and its variants. Another non-user is "the system." Beginning the user story with the phrase "As the system" may enable teams to use a waterfall approach. The system doesn't care if business value is delivered or not. With the waterfall approach, no business value is delivered until the end.  <b>Improvement:</b> Knowing the specific role for a story helps us understand the context for the story, leading to better value and focus.</p>
G	<p><b>Parakeet Value (valeur perroquet)</b>  <b>Problem:</b> Beware when the story's "so that" phrase is a restatement of the story's "I want" phrase. This points to a lack of analysis depth. The real goal of the story is not at all clear, making it easy to be off target and develop software that doesn't maximize business value. Such stories are often not analyzed well enough.  <b>Improvement:</b> remove the « So that » part and change it to real business value</p>
H	<p><b>Technic Value</b>  <b>Problem:</b> Beware of user stories where "so that" is a technical capability, not any value to the end user. This is very similar to no business value, but the value listed is technical.  <b>Improvement:</b> The improvement would be to delete this story.</p>

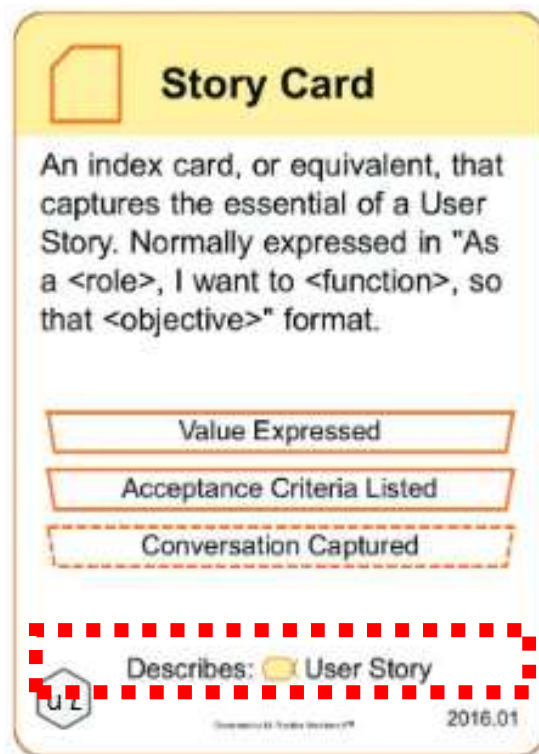


# Focus on Story Card : level of details

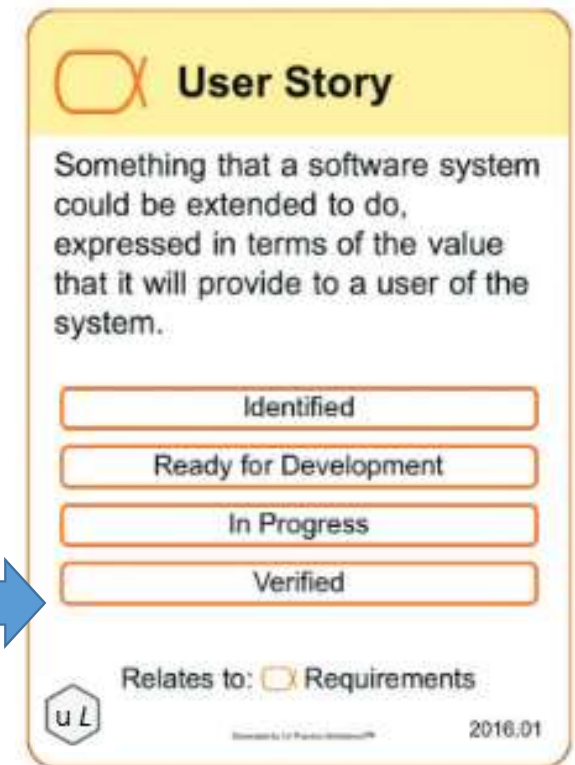


- Checkpoints (of level of details)
  - Value expressed
    - the story has been reformulated to comply with common format as described above.
  - Acceptance Criteria Listed
    - Lists the tests the story implementation must pass to be finished
  - Conversation captured (dashed => optional)
    - The development team has discussed (and recorded) and agreed on the understanding of story.

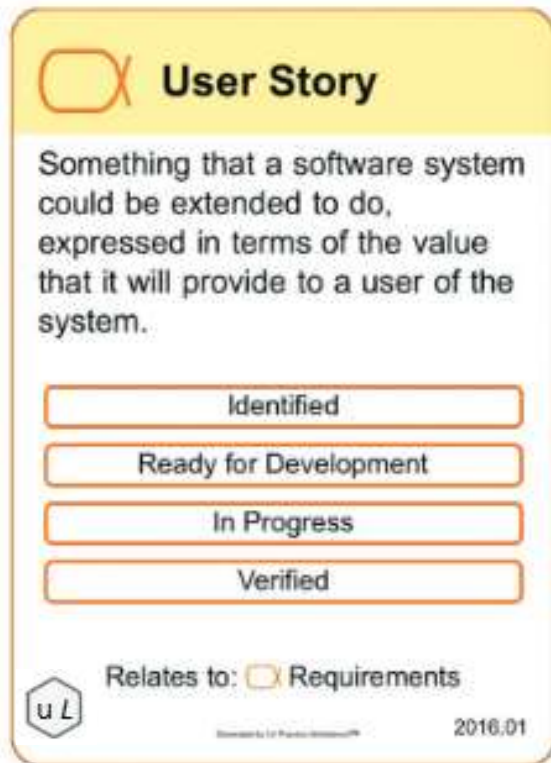
# Focus on Story Card : relations (*describes*)



- **Level of details** are about maturity of work product
- **Alpha** help to identify the health and progress of the associated element.
- **Alpha states** are about progress of an element



# User Story Lite : focus on User Story Card



- Checkpoint (of Alpha States)
  - Identified
    - **User Story is described by the customer.**
    - **User Story is prioritized by the customer.**
  - Ready for development
    - **The User Story has been broken down into tasks by the developers.**
    - **The User Story has been estimated by the developers.**
- In progress
  - **The User Story has been implemented.**
  - **The implementation has been tested.**
- Verified
  - **The Customer has approved the implementation.**

# User Story Lite : what about Test Case ?

**Prepare a User Story**

A User Story is prepared for development by discussion to build understanding and refinement of its acceptance criteria and test cases.

Entry criteria

- User Story: Identified
- Story Card: Value Expressed

Understand the Requirements

Stakeholder Representation Analysis Development Testing

- User Story: Ready for Development
- Story Card: Acceptance Criteria Listed or beyond
- Test Case: Acceptance Criteria Captured or beyond (all Test Cases)

Completion criteria

uL 2016.01

- As indicated on the card, the « Prepare a User Story » delivers « Test Case » with level of detail « Acceptance Criteria Captured ».

**Test Case**

Defines test inputs and expected results to evaluate whether a User Story is fully and correctly implemented.

Acceptance Criteria Captured

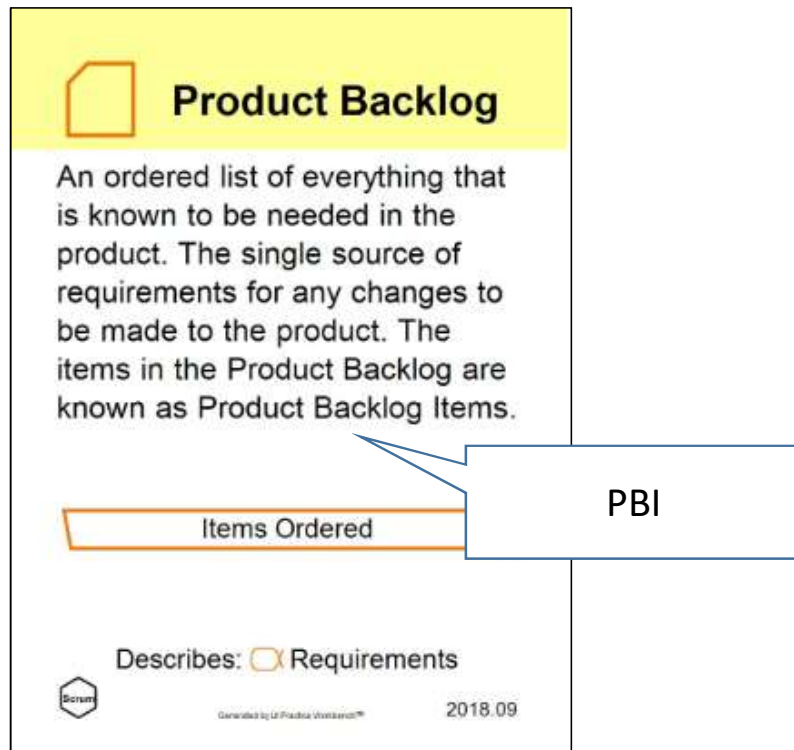
Scripted

Automated

Describes: User Story

2016.01

# Focus on Story Card: back to SCRUM



- What are Product Backlog Items ?
  - Requirements
    - User's Story : focuses on the experience — what the person using the product wants to be able to do.
    - *Traditional* requirements (focus on functionality — what the product should do)
    - Feature, Epic : composition of requirements
  - Defects
    - That involve a Bug fix
  - Tasks
    - Technical debt : "*not quite right code which we postpone making it right*"
    - Knowledge acquisition : doing research, creating prototypes, running experiments, or implementing POCs.

A

### Requirements

What the software system must do to address the opportunity and satisfy the stakeholders.

- Conceived
- Bounded
- Coherent
- Acceptable
- Addressed
- Fulfilled

### User Story

Something that a software system could be extended to do, expressed in terms of the value that it will provide to a user or system.

- Identified
- Ready for Development
- In Progress
- Verified

### Story Card

An index card, or equivalent, that captures the essential of a User Story. Normally expressed in "I, a <role>, I want to <function>, that <objective>" format.

- Value Expressed
- Acceptance Criteria Listed
- Conversation Captured

### Test Case

Defines test inputs and expected results to evaluate whether a User Story is fully and correctly implemented.

- Acceptance Criteria Captured
- Scripted
- Automated

### Splittin Stories

Small things get done in development there is a relentless drive to reduce User Stories by splitting into smaller ones. The key is to ensure that each story delivers value:

- Splits should support meaningful user interactions, no matter how small or "specialized" (think "thin end-to-end journey / slice" not technical architecture "dice"—e.g. front-end without back-end)
- Remember: each and every Test Case is a potential new story.

Kernel symbol (the mother of all practices)

verified by

things of value that a system could do. these as simple and headline descriptions on cards.

A User Story is prepared for development by discussion to build understanding and refinement of its acceptance criteria and test cases.

### Accept a User Story

The User Story implementation is evolved in close collaboration with the customers until it is acceptable to and accepted by the accepting customer representative.

Understand the Requirements

- Stakeholder Representation
- Analysis

- User Story: Identified
- Story Card: Value Expressed

Understand the Requirements

- Stakeholder Representation
- Analysis
- Development
- Testing

- User Story: Ready for Development
- Story Card: Acceptance Criteria Listed or beyond
- Test Case: Test Ideas Captured or beyond (all Cases)

Test the System

- Stakeholder Representation
- Development
- Testing

- User Story: Verified

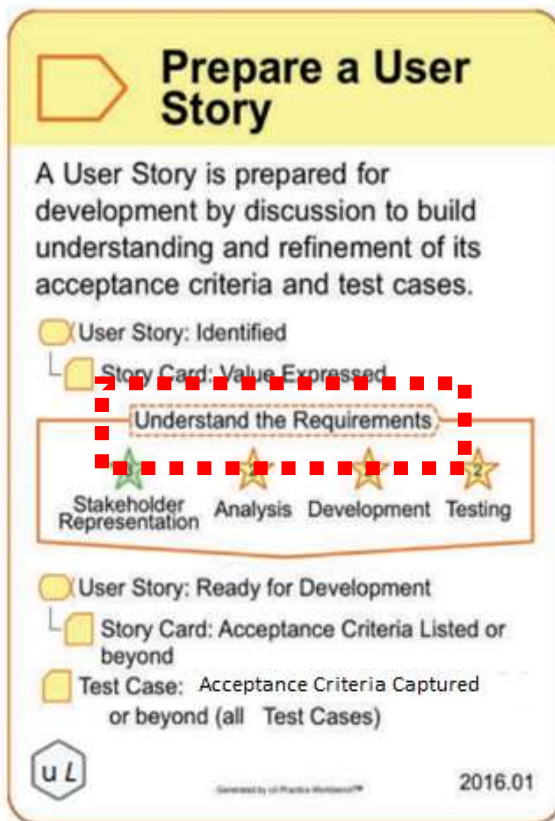


Kernel cards box

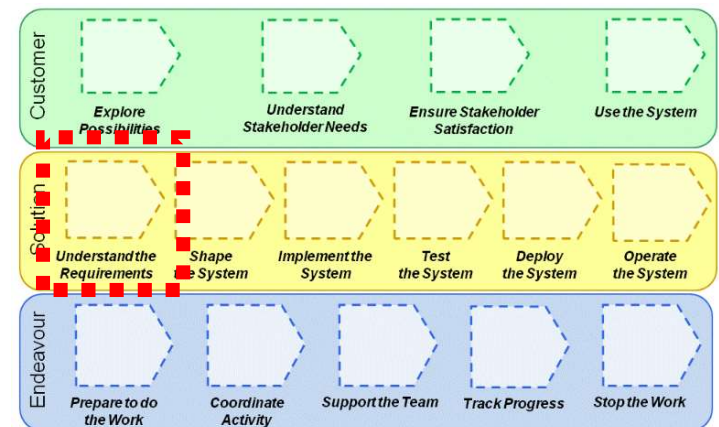
# Essence Kernel

A set of predefined elements common to all practices.

# Kernel – Activity Spaces

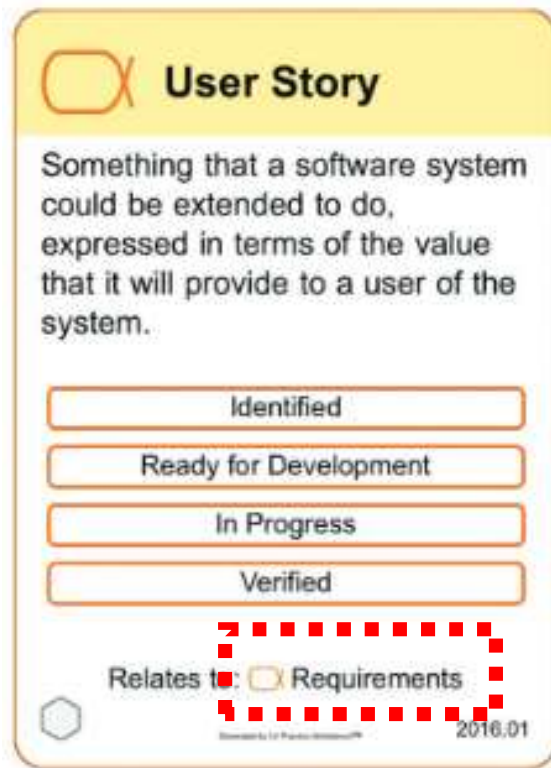


- Some elements of practice refers to elements of Kernel
  - Prepare a User Story
    - => Understands the requirements (Activity Spaces)

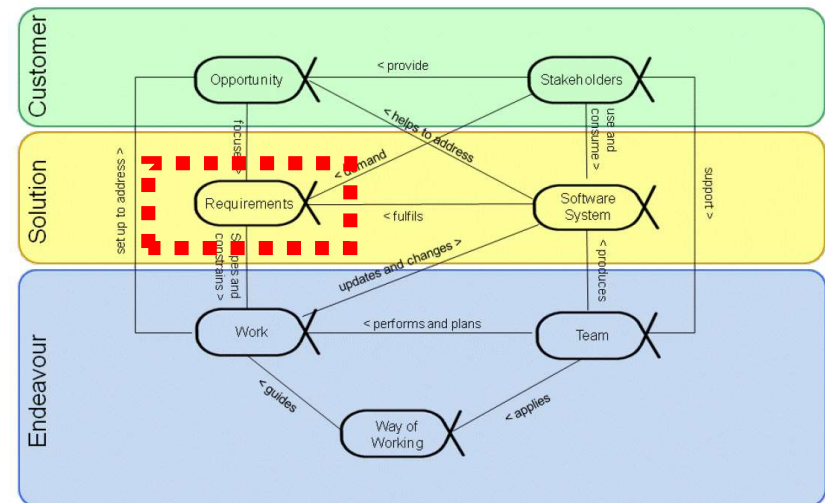




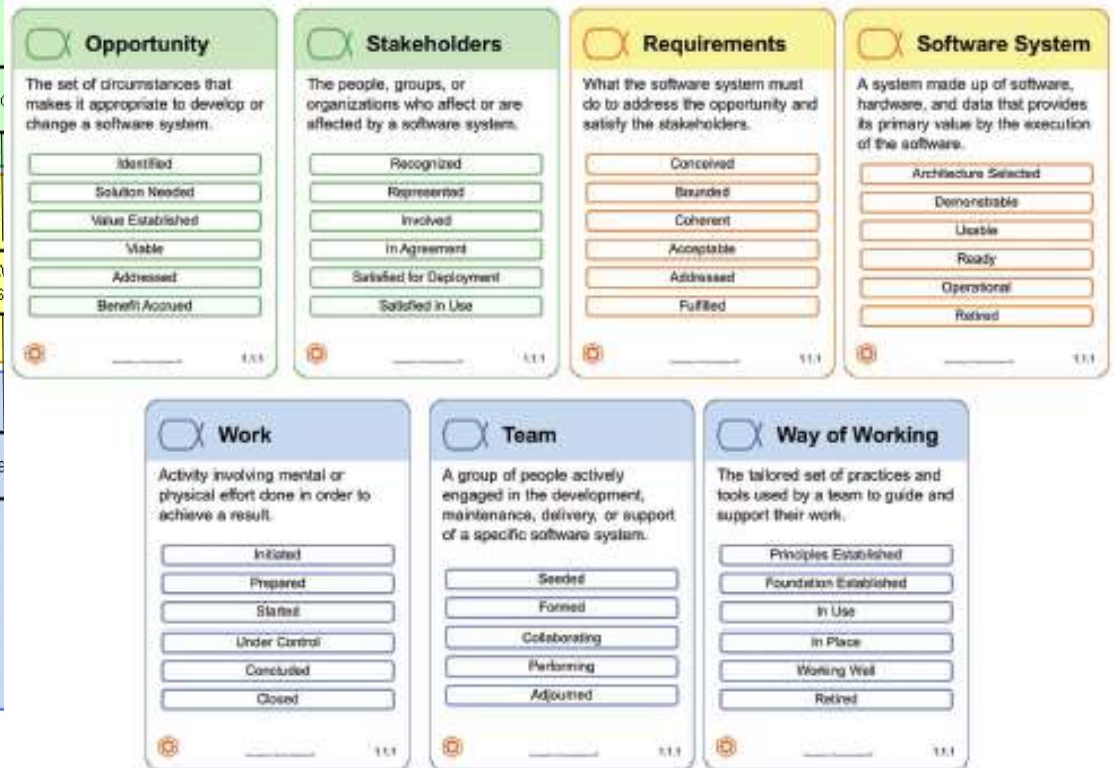
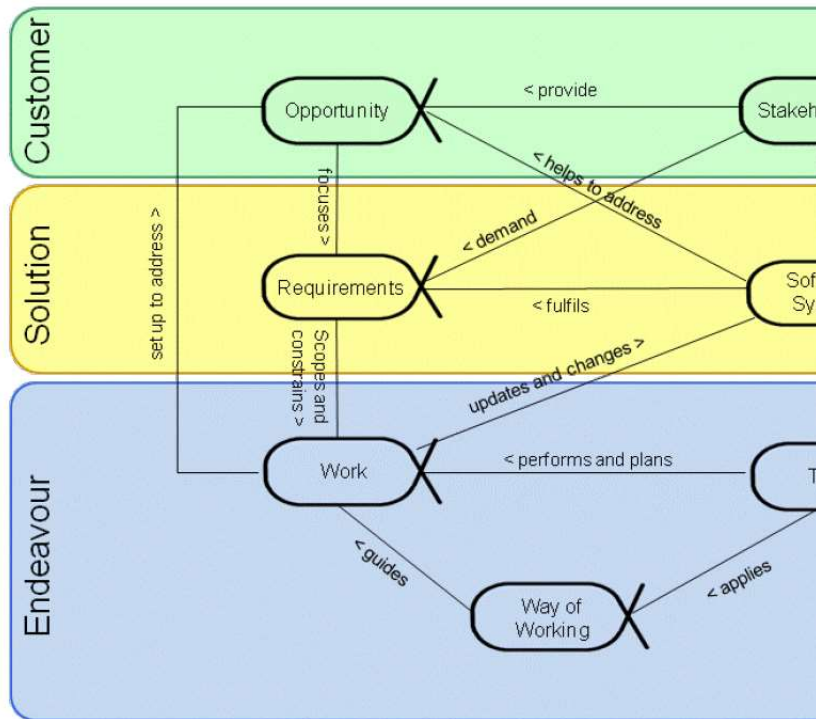
# Kernel – Alphas (1)



- Some elements of practice refers to elements in Kernel
  - User Story
    - => relates to Requirements (Alpha)



# Kernel – Alphas (2)



# Kernel - Competencies



Competency Level	Brief Description
1 - Assists	<p>Demonstrates a basic understanding of the concepts involved and can follow instructions.</p> <p>The following describe the traits of a Level 1 individual:</p> <ul style="list-style-type: none"> <li>• Understands and conducts his or her self in a professional manner.</li> <li>• Is able to correctly respond to basic questions within his or her domain.</li> <li>• Is able to perform most basic functions within the domain.</li> <li>• Can follow instructions and complete basic tasks.</li> </ul>
2 - Applies	<p>Able to apply the concepts in simple contexts by routinely applying the experience gained so far.</p> <p>The following describe the traits of a Level 2 individual:</p> <ul style="list-style-type: none"> <li>• Is able to collaborate with others within the Team.</li> <li>• Is able to satisfy routine demands and do simple work requirements.</li> <li>• Can handle simple challenges with confidence.</li> <li>• Can handle simple work requirements but needs help in handling any complications or difficulties.</li> <li>• Is able to reason about the context and draw sensible conclusions.</li> </ul>
3 - Masters	<p>Able to apply the concepts in most contexts and has the experience to work without supervision.</p> <p>The following describe the traits of a Level 3 individual:</p> <ul style="list-style-type: none"> <li>• Is able to satisfy most demands and work requirements.</li> <li>• Is able to speak the language of the competency's domain with ease and accuracy.</li> <li>• Is able to communicate and explain his or her work.</li> <li>• Is able to give and receive constructive feedback.</li> <li>• Knows the limits of his or her capability and when to call on more expert advice.</li> <li>• Works at a professional level with little or no guidance.</li> </ul>
4 - Adapts	<p>Able to apply judgment on when and how to apply the concepts to more complex contexts. Can make it possible for others to apply the concepts.</p> <p>The following describe the traits of a Level 4 individual:</p> <ul style="list-style-type: none"> <li>• Is able to satisfy complex demands and work requirements.</li> <li>• Is able to communicate with others working outside the domain.</li> <li>• Can direct and help others working within the domain.</li> <li>• Is able to adapt his or her way-of-working to work well with others, both inside and outside their domain.</li> </ul>
5 - Innovates	<p>A recognized expert, able to extend the concepts to new contexts and inspire others.</p> <p>The following describe the traits of a Level 5 individual:</p> <ul style="list-style-type: none"> <li>• Has many years of experience and is currently up to date in what is happening within the domain.</li> <li>• Is recognized as an expert by his or her peers.</li> <li>• Supports others in working on complex problems.</li> <li>• Knows when to innovate or do something different and when to follow normal procedure.</li> <li>• Develops innovative and effective solutions to the current challenges within the</li> </ul>

# Retrospective

Focus on another activity of SCRUM



# EXG2135 : Where is the Retrospective card ?



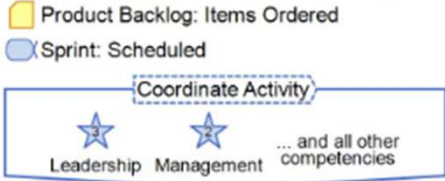
Plan and replan the work for the next 24 hours to optimize team collaboration and performance. Held daily, this is 15-minute time-boxed event for the Development Team.



- Work: Under Control
- Sprint Backlog: Forecast or beyond



Collaboratively plan the work to be performed in the Sprint and agree what can be delivered in the Sprint's Increment. A time-boxed event of no more than 8 hours for a one-month Sprint (shorter for shorter Sprints).



- Sprint: Planned
- Sprint Backlog: Forecast or beyond
- Sprint Goal: Objective Clear



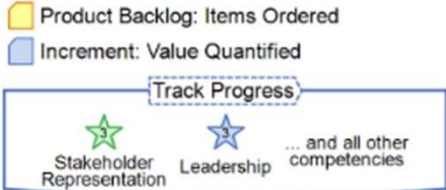
An opportunity for the Scrum Team to inspect itself and create a plan for Improvements to be enacted in the next Sprint. A time-boxed event of no more than 3 hours for a one-month Sprint (shorter for shorter Sprints).



- Improvement: Planned or beyond



An informal meeting, held at the end of the Sprint, to inspect the Increment and adapt the Product Backlog. A time-boxed event of no more than 4 hours for a one-month Sprint (shorter for shorter Sprints).



- Product Backlog: Items Ordered
- Product Backlog Item: Identified
- Sprint: Reviewed



# EXG2135 : Where is the Retrospective card ?



Plan and replan the work for the next 24 hours to optimize team collaboration and performance. Held daily, this is 15-minute time-boxed event for the Development Team.



- Work: Under Control
- Sprint Backlog: Forecast or beyond



Generated by UI Practice Workshops™

2018.09



Collaboratively plan the work to be performed in the Sprint and agree what can be delivered in the Sprint's Increment. A time-boxed event of no more than 8 hours for a one-month Sprint (shorter for shorter Sprints).

- Product Backlog: Items Ordered
- Sprint: Scheduled



- Sprint: Planned
- Sprint Backlog: Forecast or beyond
- Sprint Goal: Objective Clear



Generated by UI Practice Workshops™

2018.09



An opportunity for the Scrum Team to inspect itself and a plan for Improvements enacted in the next Sprint. A time-boxed event of no more than 3 hours for a one-month Sprint (shorter for shorter Sprints).



- Improvement: Planned or beyond



Generated by UI Practice Workshops™



## Retrospective

An informal meeting, held at the end of the Sprint, to inspect the Increment and adapt the Product Backlog. A time-boxed event of no more than 4 hours for a one-month Sprint (shorter for shorter Sprints).

- Product Backlog: Items Ordered
- Increment: Value Quantified



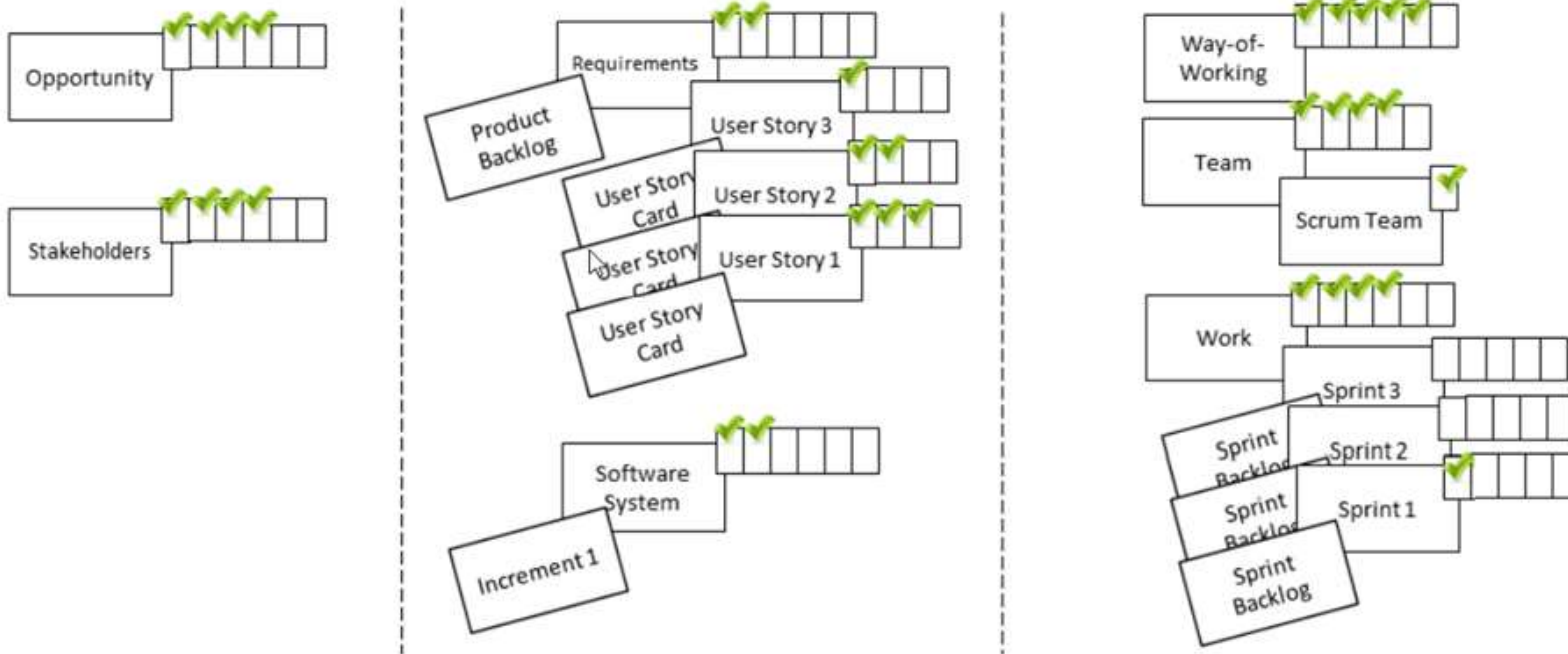
- Product Backlog: Items Ordered
- Product Backlog Item: Identified
- Sprint: Reviewed



Generated by UI Practice Workshops™

2018.09

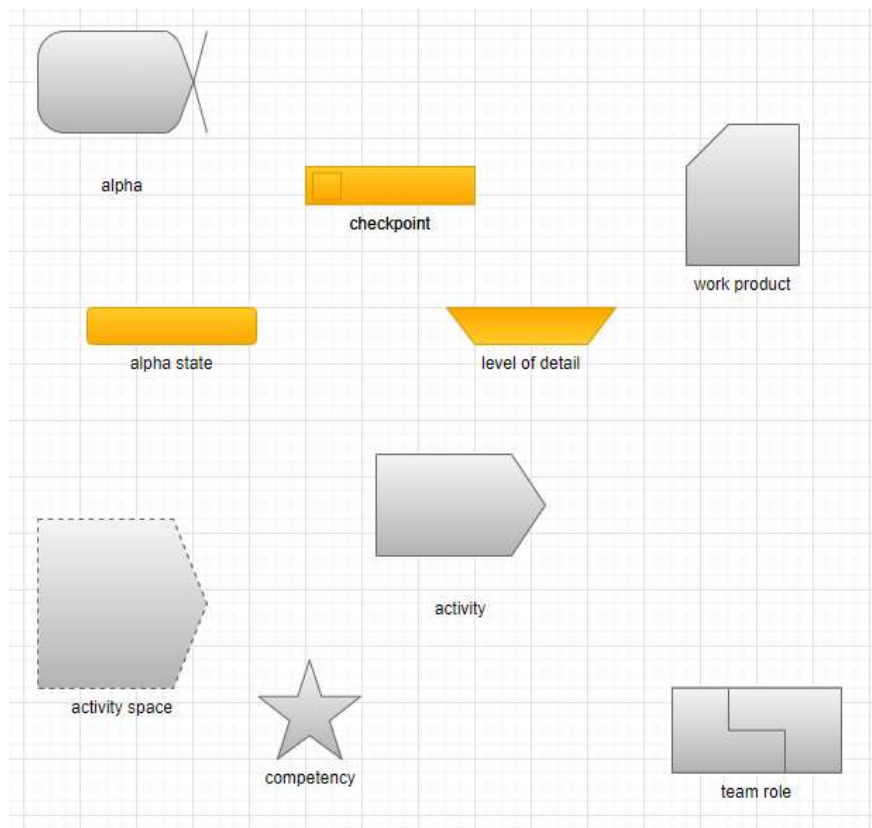
# EXA2141



# Conclusion

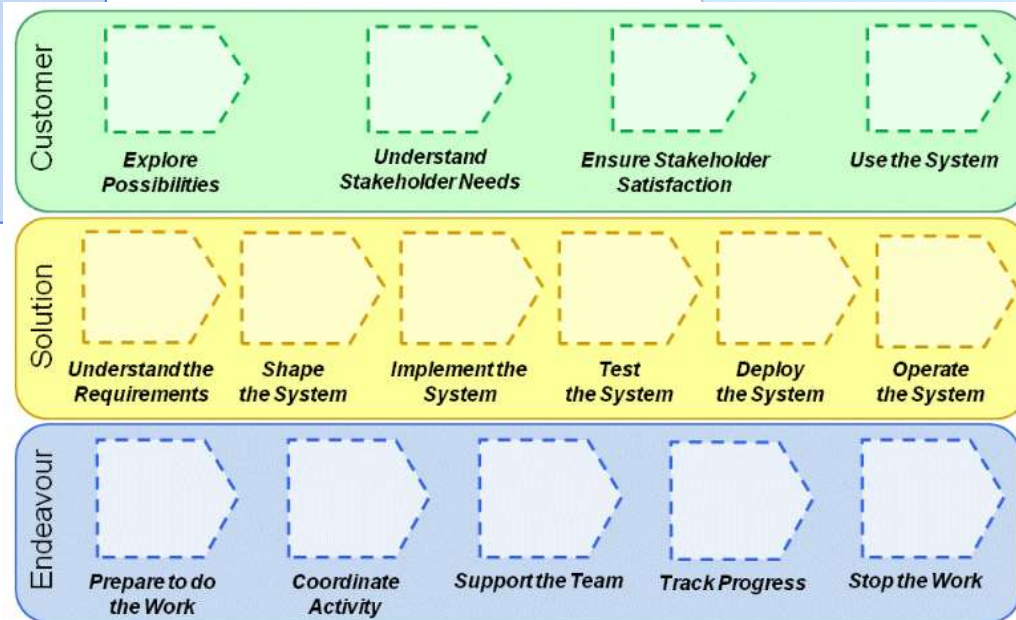
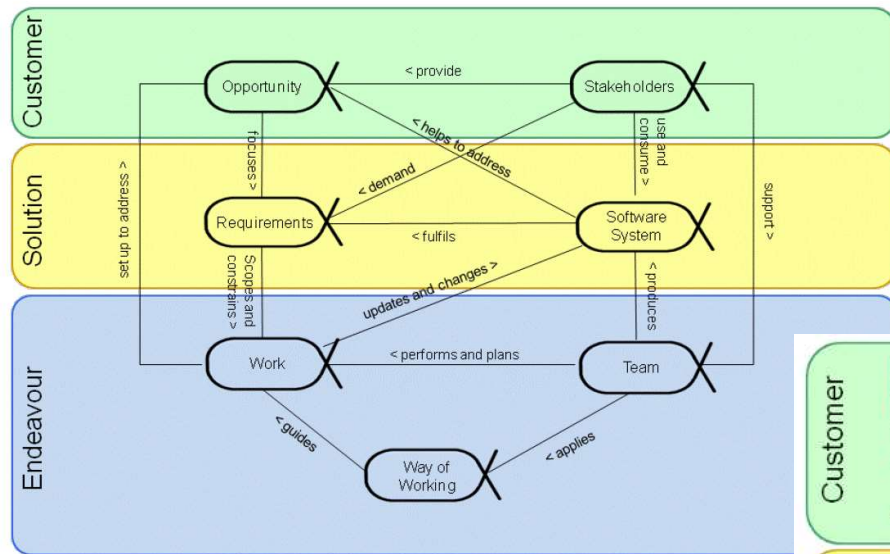


# Elements (of language) manipulated



- Role (inherited from Pattern)
- Work Product
- Activity
- Activity Spaces
- Level of Detail
- Alpha
- Alpha State
- Competency

# Elements (of kernel) manipulated



# Language, Kernel and Practice

