

# Practices and Methods: Realizing the SEMAT Vision

Ian Spence

# The Future: Practice Independence

Wolfe People

## **A Declaration of Practice Independence**

We hold these truths to be self-evident, that all practices are created equal, that they are endowed by their Creator with certain unalienable Rights, that among these are life (constant evolution), liberty (method independence) and the pursuit of excellence.

# The Future: No More Method Wars



# The Future: Better Software Development

---

WE OFFER 3 KINDS OF SERVICES  
**GOOD-CHEAP-FAST**  
BUT YOU CAN PICK ONLY TWO

---

**GOOD & CHEAP** WON'T BE **FAST**

---

**FAST & GOOD** WON'T BE **CHEAP**

---

**CHEAP & FAST** WON'T BE **GOOD**

---



# The Future: Better Teams



# The Future: It's Here Today





# Introducing the IJI Essentials

# Capturing the essence of your way-of-working

The Essence Kernel is a small practice independent process that captures the common ground that is the basis for all your work.



## A Method Foundation

Capture your practices and dynamically assemble your ways-of-working.



A Sat-Nav for your projects.  
Know where you are and where you are going.



Check progress and health.

Collect evidence in a lightweight fashion



Bringing people together to create winning teams, create one culture, and increase staff mobility

# To realize the vision we need practices

## Tactics and tricks

Local team practices captured as a set of Working Practices



## Domain Specific Practices

Extensions to the generic practices and additional domain specific practices



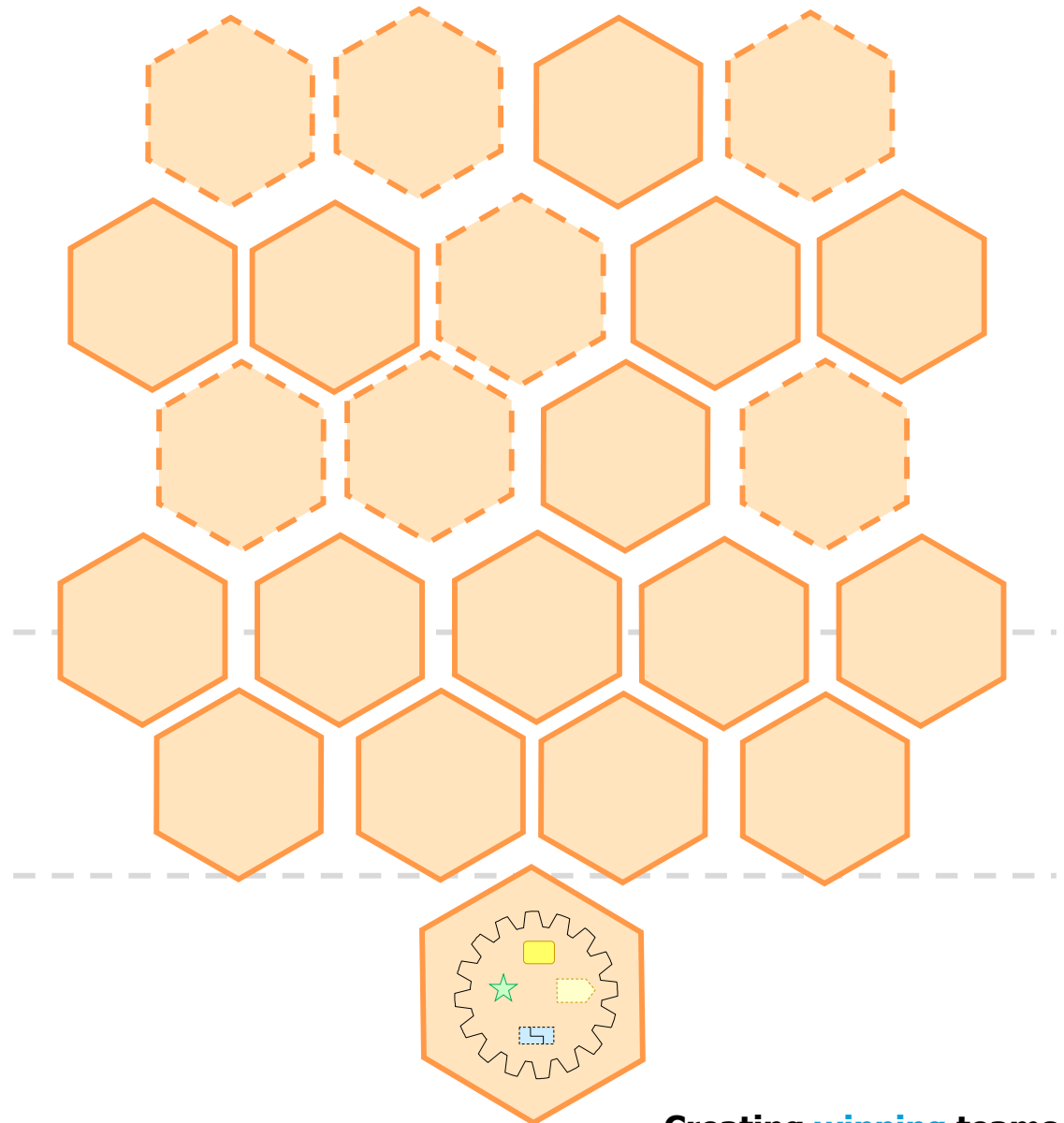
## Generic Practices

Standard practices to support effective team working and sound engineering principles.



## The Common Ground

The basic concepts applicable to all captured as a Kernel



Creating **winning** teams.

# To realize the vision we need practices

## Tactics and tricks

Local team practices captured as a set of Working Practices



## Domain Specific Practices

Extensions to the generic practices and additional domain specific practices



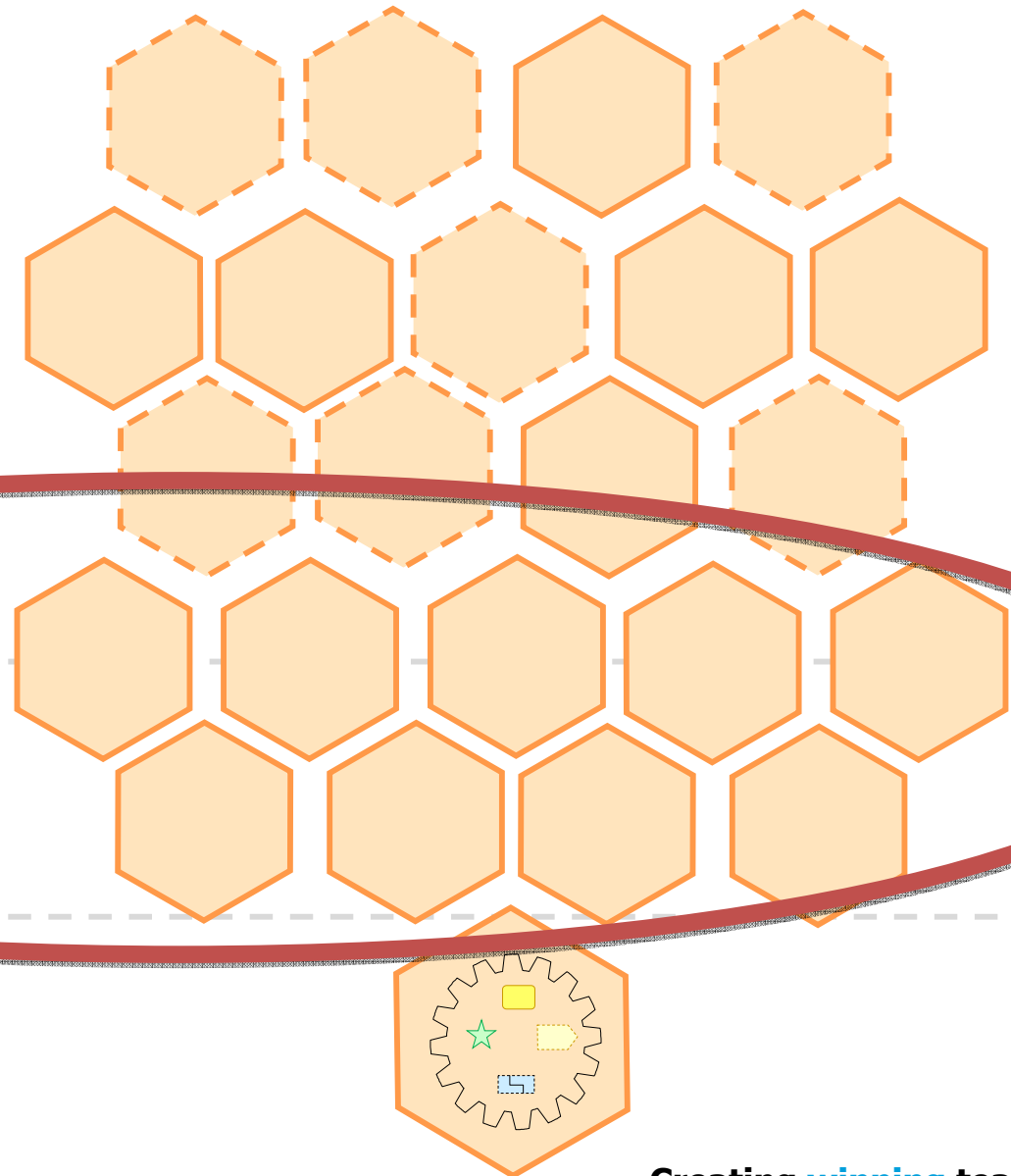
## Generic Practices

Standard practices to support effective team working and sound engineering principles.



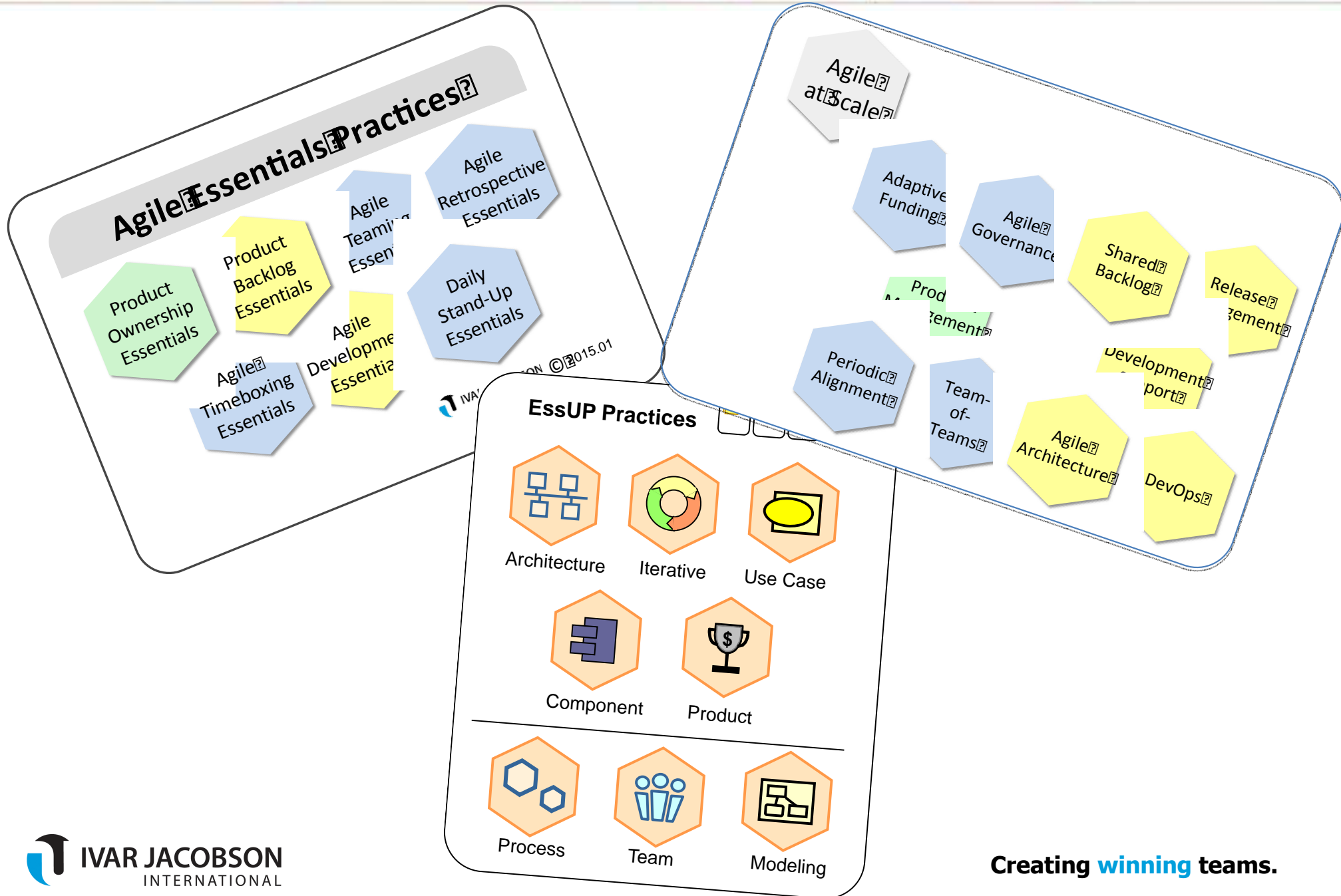
## The Common Ground

The basic concepts applicable to all captured as a Kernel



Creating **winning** teams.

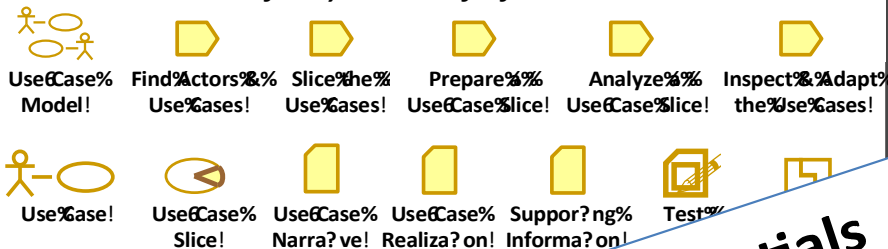
# Introducing IJI Essentials



# Introducing IJI Essentials

## Use Case 2.0

A scalable, agile practice that uses use cases to capture a set of requirements and drive the incremental development of a system to fulfil them.



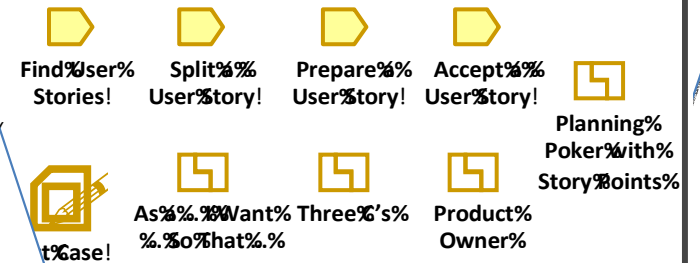
Agile at Scale

Adaptive Funding

Agile

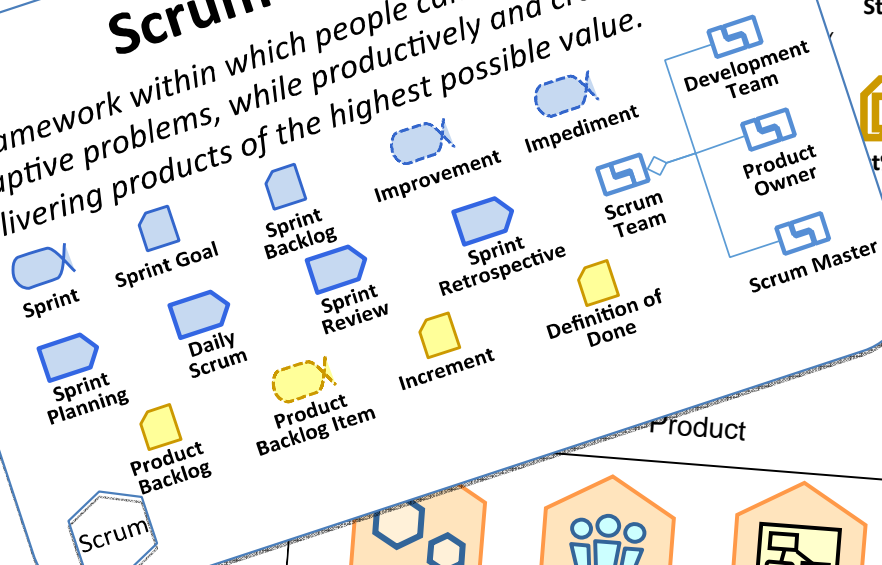
## User Story Essentials

... what the users of a software system want to do, in an informal way as part of an agile way of working.



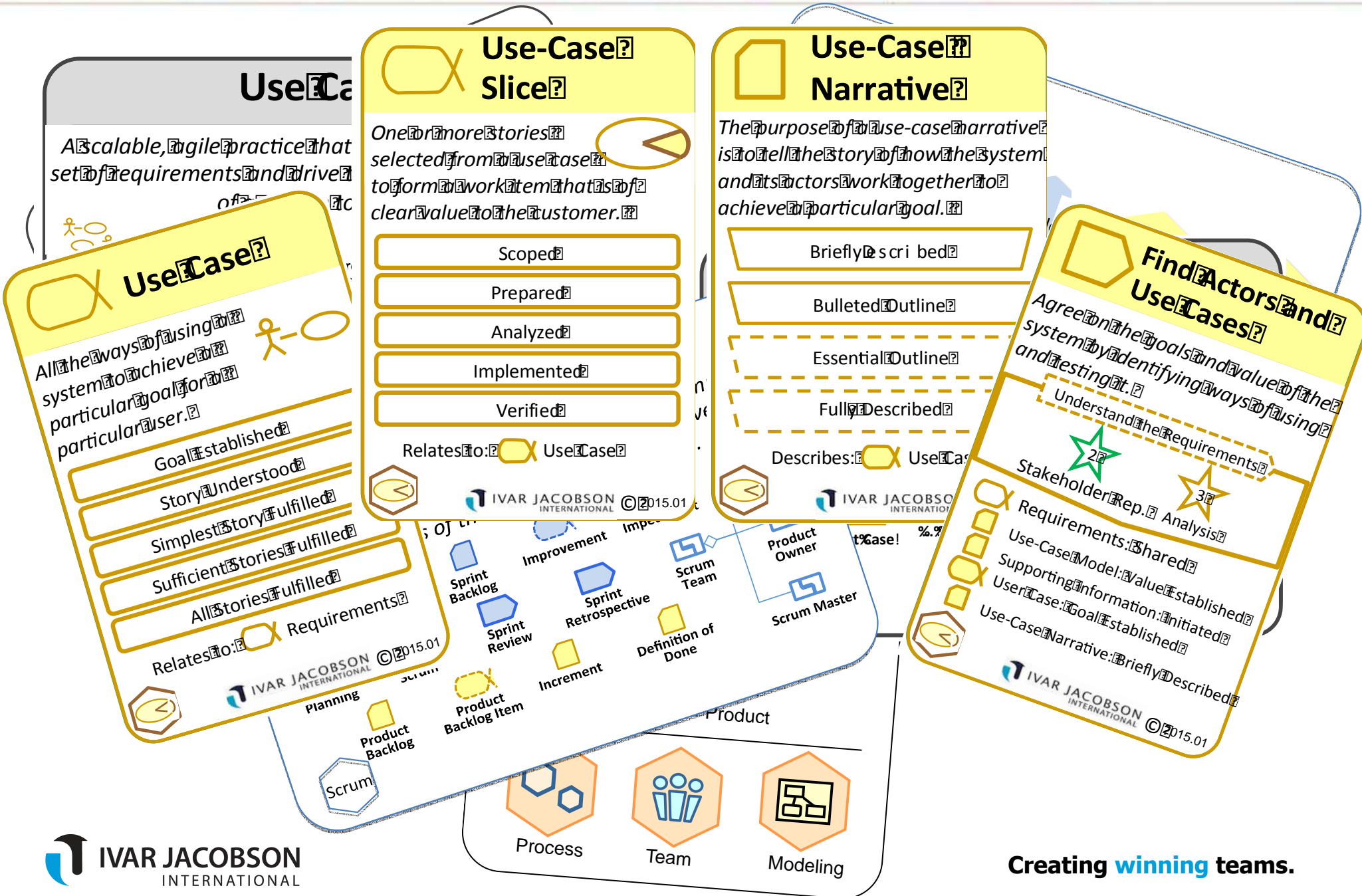
## Scrum Essentials

A framework within which people can address complex adaptive problems, while productively and creatively delivering products of the highest possible value.



IVAR JACOBSON INTERNATIONAL © 2015.01

# Introducing IJI Essentials



# Available Electronically

file:///Users/ianspence/Dropbox%20Ivar%20Jacobson%20int./Prac

Use Case 2.0 Essentials Top Sites

## Use Case 2.0 Essentials

A scalable, agile practice that uses use cases to capture a set of requirements and drive the incremental development of a system to fulfill them.

Use-Case Model	Find Actors & Use Cases	Slice the Use Cases	Prepare a Use-Case Slice	Analyze a Use-Case Slice	Inspect & Adapt the Use Cases
Use Case	Use-Case Slice	Use-Case Narrative	Use-Case Realization	Supporting Information	Test Case

(Click to navigate)

### Introduction

Use this practice to capture requirements in an accessible form and drive the development of software.

This practice allows teams to:

- Describe exactly what a software system must do
- Group parts of the requirements together
- Change the priority of what the customer wants at any time
- Produce a simple visual model and meaningful requirements that are understandable to developers and customers alike
- Take advantage of the benefits of iterative development.

### When to Apply

The Use Case 2.0 Essentials Practice should be used when the teams want to:

- Ensure customers get exactly what they want, in the order they want it and when they want it
- Deliver tested software early



# Available Electronically

The screenshot shows a web browser window displaying the 'Use Case 2.0 Essentials' page. The page has a left sidebar with a navigation menu and a main content area. The main content area includes a title 'Use Case 2.0 Essentials', a description, a row of five yellow arrow icons with labels, and a section titled 'Introduction' with a bulleted list.

**Callout 1:** The raw practice. Also directly accessible from the practice overview page.

**Callout 2:** The practice in the context of the kernel, which is as it would appear when composed in a method.

**Callout 3:** Direct access to supporting resources such as the cards, examples, references and templates

**Callout 4:** (Click to navigate)

**Navigation Menu (Left Sidebar):**

- Use Case 2.0 Essentials
  - Use Case
  - Use-Case Slice
  - Supporting Information
  - Test Case
  - Use-Case Model
  - Use-Case Narrative
  - Use-Case Realization
  - Analyze a Use-Case Slice
  - Find Actors and Use Cases
  - Inspect and Adapt the Use Cases
  - Prepare a Use-Case Slice
  - Slice the Use Cases
- Use Case 2.0 In Context
  - Things to Work With
  - Things to Do
  - Competencies
  - Patterns
  - Areas of Concern
- Resources
  - Cards
  - Examples
  - Glossary of Terms
  - References
  - Slate Progression Table
  - Templates
  - Tools
  - Other
- FAQ
- Key
- Notices

**Main Content Area:**

**Use Case 2.0 Essentials**

A scalable, agile practice that uses use cases to... a set of requirements and drive the incremental development of a system to fulfill them.

**Navigation Row:** Use-Case Model, Find Actors & Use Cases, Slice the Use Cases, Prepare a Use-Case Slice, Analyze a Use-Case Slice, Inspect & Adapt the Use Cases

**Introduction**

Use this practice to...

This practice allows teams to...

- Describe exactly what...
- Group parts of the requirements...
- Change the priority of what...
- Produce a simple visual model and meaningful requirements that are understandable to developers and customers alike

All generated from Practice Workbench for individual practices and methods composed from sets of practices.

# Available Electronically

The screenshot shows a web browser window with the address bar displaying a file path. The page title is "Use Case 2.0 Essentials". On the left, there is a navigation menu with a tree view containing items like "Use Case 2.0 Essentials", "Use Case", "Use-Case Slice", "Supporting Information", "Test Case", "Use-Case Model", "Use-Case Narrative", "Use-Case Realization", "Analyze a Use-Case Slice", "Find Actors and Use Cases", "Inspect and Adapt the Use Cases", "Prepare a Use-Case Slice", "Slice the Use Cases", "Use Case 2.0 In Context", "Things to Work With", "Things to Do", "Competencies", "Patterns", "Areas of Concern", "Resources", "Cards", "Examples", "Glossary of Terms", "References", "State Progression Table", "Templates", "Tools", "Other", "FAQ", "Key", and "Notices".

The main content area has a yellow header with a "Use Case" icon and title. Below it is a flowchart with five boxes connected by downward arrows: "Goal Established", "Story Structure Understood", "Simplest Story Fulfilled", "Sufficient Stories Fulfilled", and "All Stories Fulfilled". Below the flowchart is a link "(Click to navigate)".

To the right of the flowchart, there is text defining a use case and a list of characteristics:

A use case is a goal for a particular user.

A use case is:

- A sequence of actions a system performs that yields an observable result of value to a particular user.
- That specific behaviour of a system which participates in a collaboration with a user.
- The smallest unit of work for the user.
- The context of the user.

A use case is self-consistent state.

Taken together, the stories cover both how to use the system. To understand the use case, we tell stories. The stories cover both how to achieve the goal and how to handle any problems that arise on the way. They help us to understand the use case and implement it slice by slice.

At the bottom, there is a table with the following structure:

<b>Contains:</b>	<b>Described by:</b>	<b>Related to:</b>
<ul style="list-style-type: none"> <li>• 1..N <a href="#">Use-Case Slice</a></li> </ul>	<ul style="list-style-type: none"> <li>• 1 <a href="#">Use-Case Narrative</a></li> <li>• 1..N <a href="#">Use-Case Realization</a></li> <li>• 1..N <a href="#">Test Cases</a></li> </ul>	<ul style="list-style-type: none"> <li>• <a href="#">Requirements</a> (Parent)</li> </ul>
<b>Properties:</b>	<b>Patterns:</b>	<b>Resources:</b>
N/A	N/A	<a href="#">Additional Guidelines</a>
<b>Progressed by:</b>		
<ul style="list-style-type: none"> <li>• <a href="#">Analyze a Use-Case Slice</a></li> <li>• <a href="#">Find Actors and Use Cases</a></li> </ul>		

All the information about an element on one simple, scrollable page.

Uses the language to demonstrate the relationship between elements and aid navigation.



# Available Electronically

The screenshot shows a web browser window with the address bar containing a file path. The page title is "Use Case 2.0 Essentials". The left sidebar contains a navigation menu with the following items:

- Use Case 2.0 Essentials
  - Use Case
  - Use-Case Slice
  - Supporting Information
  - Test Case
  - Use-Case Model
  - Use-Case Narrative
  - Use-Case Realization
  - Analyze a Use-Case Slice
  - Find Actors and Use Cases
  - Inspect and Adapt the Use Cases
  - Prepare a Use-Case Slice
  - Slice the Use Cases
- Use Case 2.0 In Context
  - Things to Work With
  - Things to Do
  - Competencies
  - Patterns
- Areas of Concern
- Resources
  - Cards
  - Examples
  - Glossary of Terms
  - References
  - State Progression Table
  - Templates
  - Tools
  - Other
- FAQ
- Key
- Notices

The main content area is titled "Slice the Use Cases" and contains a checklist under the heading "States". The checklist items are:

- Goal Established:** Firstly the goal of the use case has been established.
- Primary actors named:** The use case's primary actors have been identified and named.
- Goal and value clearly described:** The goal of the use case and the value it will provide have been clearly described (i.e. the use-case narrative is detailed towards Briefly Described).
- Stakeholders agree upon the goal:** All relevant stakeholders agree upon the goal the use case must fulfill.
- In or out of scope?:** It is clear whether the use case is in or out of scope
- Use-case narrative is 'Briefly Described':** The Use-case narrative is [Briefly Described](#) or beyond.

Progressed by: [Find Actors and Use Cases](#) and [Inspect and Adapt the Use Cases](#)

- Story Structure Understood:** The structure of the use-case narrative has been understood enough for the team to start work identifying and implementing the first use-case slices.
- Basic flow determined:** The main steps of the basic flow have been determined.
- Nature of other flows determined:** The type and nature of the other flows that make up the use case have been determined.
- Start and end are clear:** The start and end of the use case are clear.
- Most common stories identified:** The most common stories have been identified.
- Most important slices are 'Prepared':** The most important Use-Case Slices (those that cover the essential behavior of the use case) are in the [Prepared](#) state.

Progressed by: [Prepare a Use-Case Slice](#) and [Slice the Use Cases](#)

- Simplest Story Fulfilled:** The Software System fulfills the simplest story that allows a user to achieve the goal.

Checklist presented in an easily accessible and printable format.



# Available Electronically

file:///Users/ianspence/Dropbox%20(Ivar%20Jacobson%20Int.)/Prac...

Use Case 2.0 Essentials

<input type="checkbox"/> All use-case slices are 'Verified'	All the use cas
---	-----------------

Progressed by: [Analyze a Use-Case Slice](#), [Inspect and Adapt the Use Cases](#), [Slice the Use Cases](#)

### Additional Guidelines

#### Review Guidelines

##### Essential Qualities

1. The use case has a descriptive name that represents the primary goal of the user (more properly, the actor).
2. The use case has an active name consisting of a verb followed by at least one noun.
3. The use case has an actor that starts the use case.
4. The use case produces a meaningful and observable result.
5. The use case has a brief description that summarizes the value that it provides.
6. The stakeholders agree that the use case adds value to the system.
7. The use case is self-contained and always leaves the system in a consistent state.
8. The use case is independently verifiable.
9. Its actors represents users, humans or systems, each of which truly plays a role in the use case.

##### Evaluation Methods

The purpose of evaluating a use case is to make sure that all stakeholders agree on its:

- Value; do they want what the use case provides?
- Importance; do they really need it?
- Feasibility; is it achievable within the constraints of the implementing system?
- Timing; when is it needed?

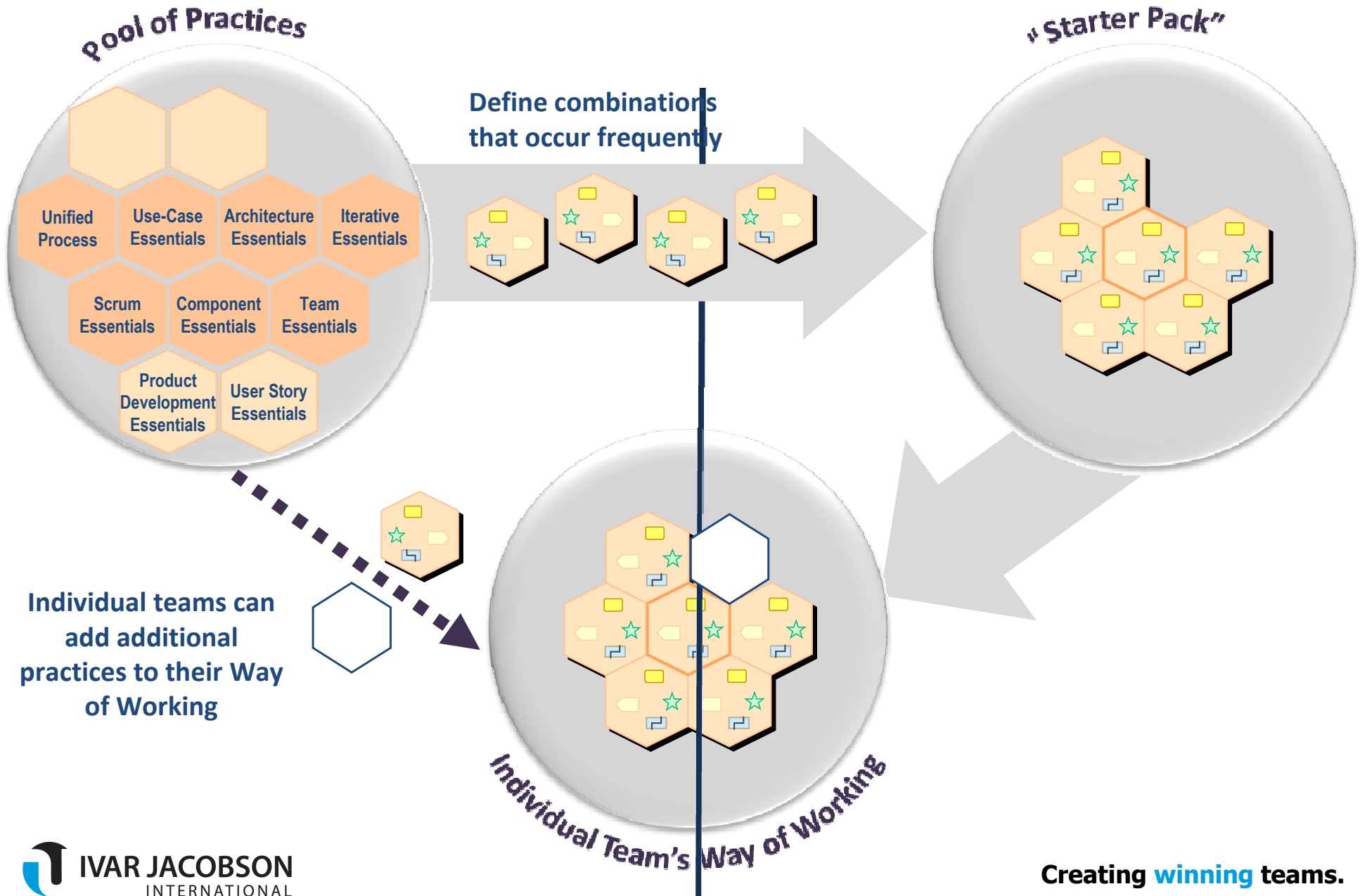
##### Review consistency

Review the use case in order to ensure consistency between it and the other use cases. Consider the following checkpoints when identifying or evaluating a use case:

- All of the essential qualities are in place.
- It has been evolved to a suitable state.
- It has been detailed appropriately for its state and the type of system being built.
- It has been prioritized relative to the other use cases.
- It has been estimated relative to the other use cases.
- It and the value it provides are clearly distinct from the other use cases.

Additional embedded guidance can be provided as part of the practice but is not obligatory.

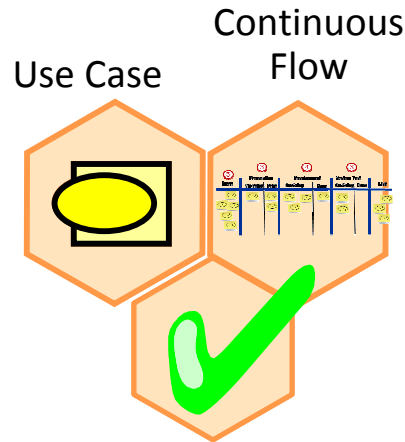
# IJI Essentials: Liberating the Practices



# Mix and Match Practices to Empower Your Teams

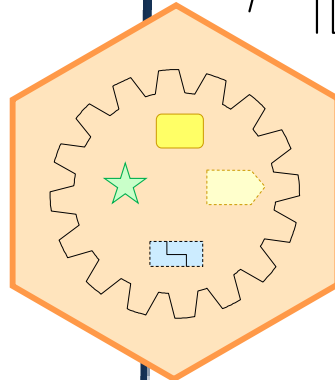


Team B

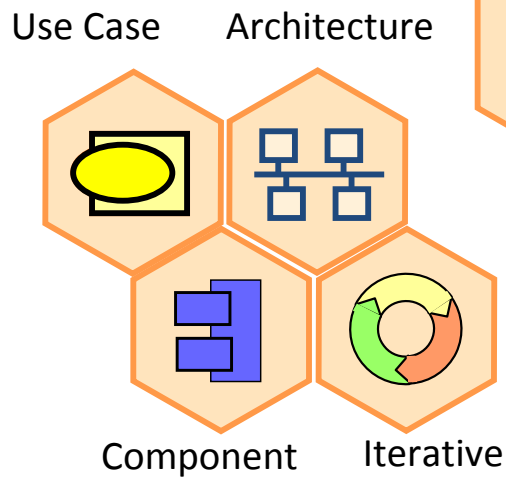


Agile Development

Team's build their way-of-working on top of pre-defined common ground and in-line with organizational principles and policies.

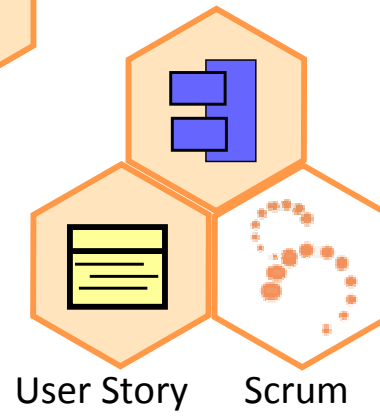


Team A



Component Iterative

Component



User Story Scrum



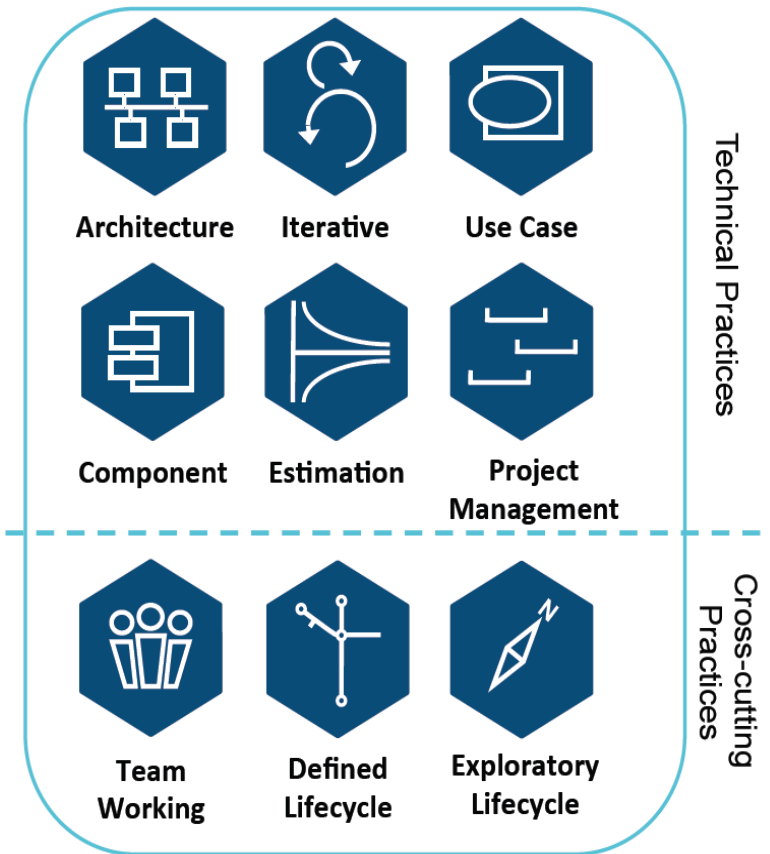
Team C

Practices enable teams to work the way that works for them

# Example: Rule Financial



## Agile Building Blocks



## Project: *rulebook*<sup>TM</sup> - Defined



assembled into

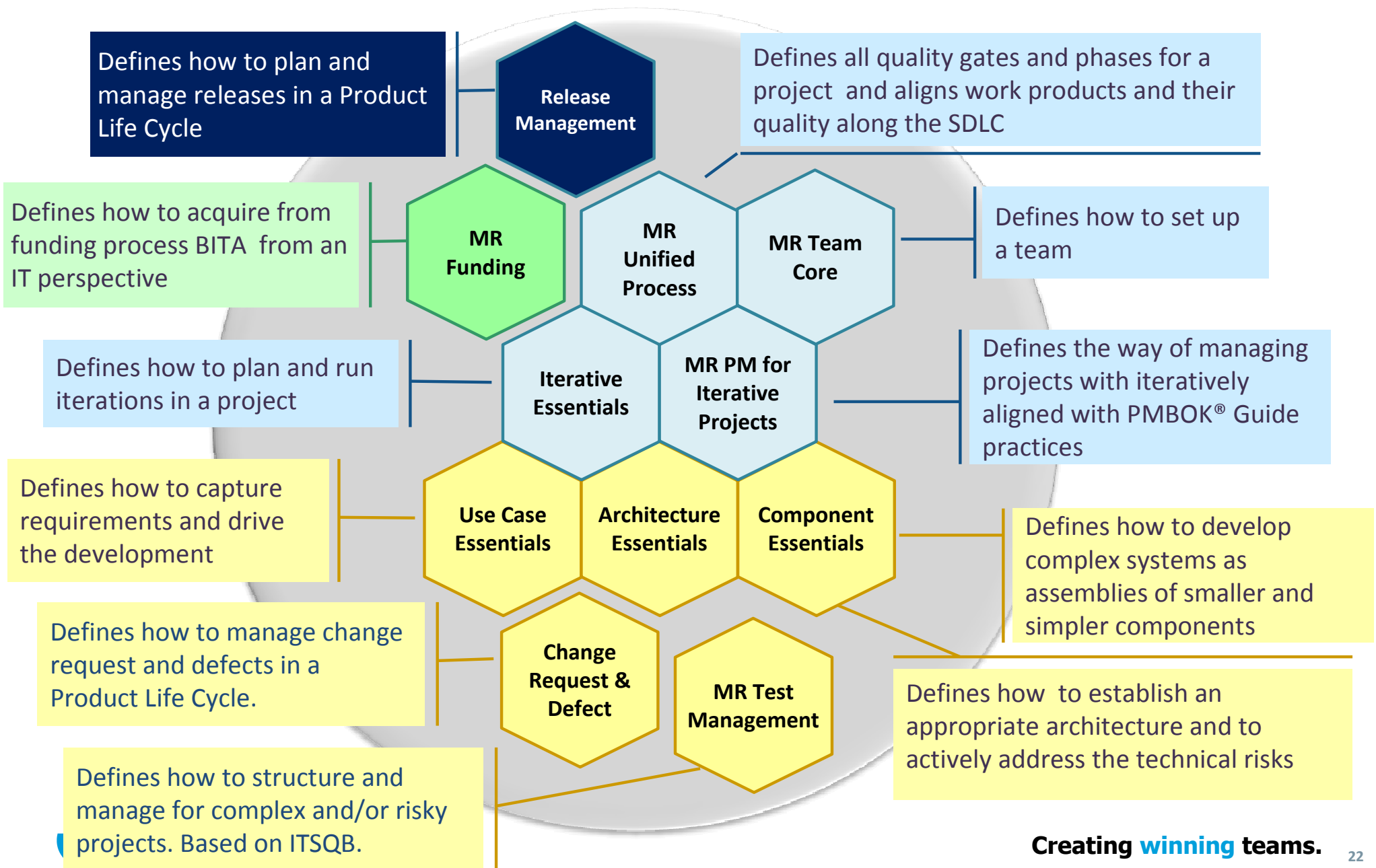
## Project: *rulebook*<sup>TM</sup> - Exploratory



assembled into

# Practices @ Rule Financial

# Example: MunichRe





It's a world of practices – and it's not standing still.



It's a world of practices – and it's not standing still.

The Kernel the great enabler.



It's a world of practices – and it's not standing still.

Practices the great communicator.



# Competitive

## IJI Essentials

Helping you keep pace  
with your technology and  
your business.

# Adv

customers system downstream members position leader choice acquire position leader performing procedures equity value-creating consumer target created lower business segment product firm-specific including examples develop create upstream perform embedded following Distinctive identified difficult sustain Partner Well-served available innovation achieve competencies way exceed one useful overall structure known differentiated diagram costs disappear capability sustainable much simply operates creating

# Questions?

## North America Office

+1 703 338 5421  
info-  
usa@ivarjacobson.com

## Swedish Office

+46 8 515 10 174  
info-sweden@ivarjacobson.com



## UK Office

+44 (0)1189 001 460  
info-eur@ivarjacobson.com

## Asia Office

+8610 824 86030  
info-asia@ivarjacobson.com

[www.ivarjacobson.com](http://www.ivarjacobson.com)