Practices and Methods: Realizing the SEMAT Vision

Ian Spence
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.
Creating winning teams.

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

Check progress and health.
Collect evidence in a lightweight fashion.

A Method Foundation

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

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
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.
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 fulfill them.

Scrum Essentials

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

User Story Essentials

Express what the users of a software system want it to do in an informal way as part of an agile way of working.
Introducing IJI Essentials

**Use Case**
All the ways of using a system to achieve a particular goal for a particular user.
- Goal Established
- Story Understood
- Simplest Story Fulfilled
- Sufficient Stories Fulfilled
- All Stories Fulfilled

Relates to: **Requirements**

**Use-Case Slice**
One or more stories selected from a use case to form a work item that is of clear value to the customer.
- Scoped
- Prepared
- Analyzed
- Implemented
- Verified

Relates to: **Use Case**

**Use-Case Narrative**
The purpose of a use-case narrative is to tell the story of how the system and its actors work together to achieve a particular goal.
- Briefly Described
- Bulleted Outline
- Essential Outline
- Fully Described

Describes: **Use Case**

**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

<table>
<thead>
<tr>
<th>Stakeholder Rep. Analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Requirements:</strong> Shared</td>
</tr>
<tr>
<td><strong>Use-Case Model:</strong> Value Established</td>
</tr>
<tr>
<td><strong>Supporting information:</strong> Initiated</td>
</tr>
<tr>
<td><strong>User Case:</strong> Goal Established</td>
</tr>
<tr>
<td><strong>Use-Case Narrative:</strong> Briefly Described</td>
</tr>
</tbody>
</table>

Creating winning teams.
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.

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
- Ensure that the user software will work early in the project
Available Electronically

The raw practice. Also directly accessible from the practice overview page.

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

Direct access to supporting resources such as the cards, examples, references and templates

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

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

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

Checklist presented in an easily accessible and printable format.

<table>
<thead>
<tr>
<th>Goal Established:</th>
<th>Firstly the goal of the use case must be established.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Primary actors named</td>
<td>The use case's primary actors have been identified and named.</td>
</tr>
<tr>
<td>Goal and value clearly described</td>
<td>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).</td>
</tr>
<tr>
<td>Stakeholders agree upon the goal</td>
<td>All relevant stakeholders agree upon the goal the use case must fulfill.</td>
</tr>
<tr>
<td>In or out of scope?</td>
<td>It is clear whether the use case is in or out of scope</td>
</tr>
<tr>
<td>Use-case narrative is 'Briefly Described'</td>
<td>The Use-case narrative is Briefly Described or beyond.</td>
</tr>
</tbody>
</table>

**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
- Nature of other flows determined
- Start and end are clear
- Most common stories identified
- Most important slices are 'Prepared'

**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.
Additional embedded guidance can be provided as part of the practice but is not obligatory.
IJI Essentials: Liberating the Practices

Define combinations that occur frequently

Individual teams can add additional practices to their Way of Working

Creating winning teams.
Mix and Match Practices to Empower Your Teams

Practices enable teams to work the way that works for them.

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

Creating winning teams.
Example: Rule Financial

Agile Building Blocks

- Architecture
- Iterative
- Use Case
- Component
- Estimation
- Project Management

Technical Practices

Cross-cutting Practices

Project: rulebook™ - Defined

assembled into

Project: rulebook™ - Exploratory

assembled into

Practices @ Rule Financial

Creating winning teams.
Creating winning teams.

Example: MunichRe

- Defines how to plan and manage releases in a Product Life Cycle
- Defines how to acquire from funding process BITA from an IT perspective
- Defines how to plan and run iterations in a project
- Defines how to capture requirements and drive the development
- Defines how to manage change request and defects in a Product Life Cycle
- Defines how to structure and manage for complex and/or risky projects. Based on ITSQB.

- Defines all quality gates and phases for a project and aligns work products and their quality along the SDLC
- Defines how to set up a team
- Defines the way of managing projects with iteratively aligned with PMBOK® Guide practices
- Defines how to develop complex systems as assemblies of smaller and simpler components
- Defines how to establish an appropriate architecture and to actively address the technical risks

MR Funding
MR Unified Process
MR Team Core
Release Management
Iterative Essentials
MR PM for Iterative Projects
Use Case Essentials
Architecture Essentials
Component Essentials
Change Request & Defect
MR Test Management

Creating winning teams.
Creating winning teams.
It’s a world of practices – and it’s not standing still.
Creating winning teams.

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.
Creating winning teams.

IJII Essentials
Helping you keep pace with your technology and your business.
Questions?