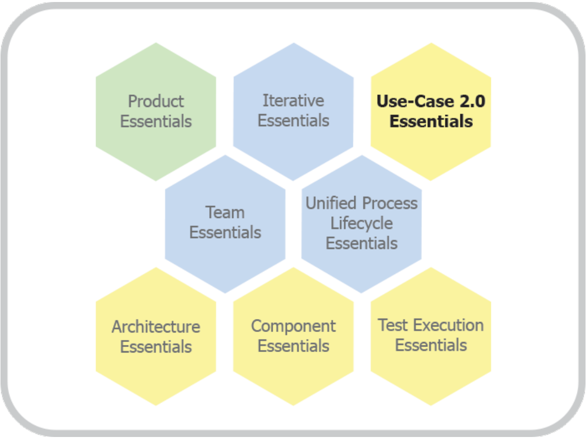
IVAR JACOBSON
INTERNATIONAL

Use-Case 2.0 Essentials

Part of the IJI Essential Unified Process Practice Pack



The Essential Unified Process (EssUP) focuses on the essentials to provide a pre-built assembly of eight easy-to-use practices that can be mixed and matched and used in different circumstances – all of them compatible with agile values and thinking. Use-case driven, iterative, component-based and architecturally-centric the practices provide an extensible framework for the addition of further practices.

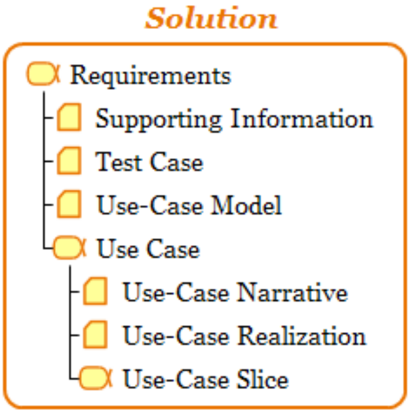
Practice Overview

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 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.

Things to Work With

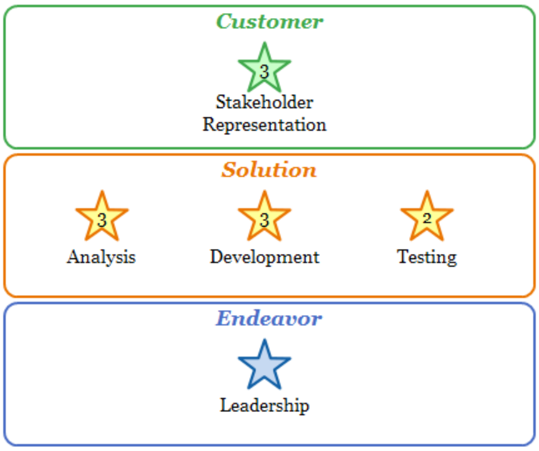


This practice involves the production of a number of requirements, design and test elements:

- A use case based specification of the requirements, stories and test cases.
- The realization of the use cases to drive the development of the software.

Creating winning teams.

Key Competencies



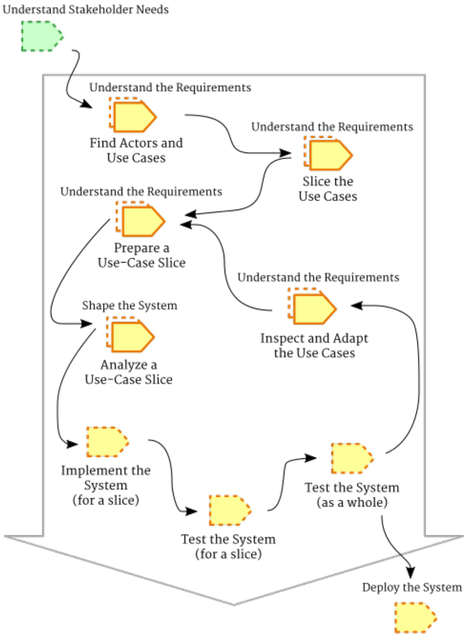
This practice requires the team to be skilled in software requirements capture, design, coding, integration and testing.

The most important skills are those of Stakeholder Representation and Analysis as, without these, the scope for development may be incorrectly set or the use-case slices not prioritized according to business needs.

Things to Do

The practice starts by finding actors and use cases, and selecting and prioritizing the parts (slices) of the use cases to be developed.

It continues by detailing the use-case slices and, more importantly, the test cases required to verify each slice. Focus then switches to implementing software in order to pass the test cases. The practice concludes by tracking progress in terms of verified, working software and feeding back the outcomes in order to handle change and better support the team.


Understand Stakeholder Needs
Understand the Requirements
Find Actors and Use Cases
Understand the Requirements
Slice the Use Cases
Understand the Requirements
Prepare a Use-Case Slice
Understand the Requirements
Inspect and Adapt the Use Cases
Shape the System
Analyze a Use-Case Slice
Implement the System (for a slice)
Test the System (for a slice)
Test the System (as a whole)
Deploy the System

Creating winning teams.