August 02, 2006

The Essential Unified Process

A fresh start for process

(Page 1 of 6)

Ivar Jacobson, Pan Wei Ng, and Ian Spence

The Essential Unified Process integrates practices from the unified process camp, agile methods camp, and process improvement camp.

Ivar is one of the "fathers" of components and component architecture, use cases, UML, and the Rational Unified Process. Pan-Wei and Ian are chief scientists at Ivar Jacobson Consulting. The authors can be contacted at http://www.ivarjacobson.com.

Everybody recognizes the need for processes to improve how software is developed. Everybody recognizes the need for agility, flexibility, and adaptability. And everybody agrees on the need for quality. But many of us find that existing processes are cumbersome, constraining, and a hindrance to our creative juices.

Developers are tired of process. The Unified Process has become too heavy, the process improvement programs require too much boring work, and the agile camp promises too much. Still, we know we need good practices to develop good software on time and on budget. In short, we need to fundamentally reengineer the way we design, configure, teach, adopt, and deploy process.

The Essential Unified Process

The Essential Unified Process (EssUP) is a new process that stands on the shoulders of modern software development practices. EssUP (www.ivarjacobson.com) is a fresh start that cautiously integrates successful practices from the unified process camp, the agile methods camp, and the process improvement camp, each contributing different capabilities.

Among the reasons we need a new process are:

- Traditional software processes are too heavy. No one reads large and lengthy process descriptions.
- Process must focus to support developers, not just process experts.

- Process must assist teams to get product quality as well as process quality; thus not only pass CMMI appraisals, but also deliver good software. The focus of any software development process must be on producing good software.
- Process must provide agility with discipline, balancing the need for governance without stifling creativity.
- The approach must let project teams (developers without the help of process engineers) easily add good practices on top of existing processes.
- Process should empower the team.

New Paradigm: Practices as First-Class Citizens

Traditional processes (such as the Unified Process) are described by their different activities and artifacts. These activities and artifacts may serve different purposes—doing requirements with use cases, test-driven design, or component-based development. These practices are not explicit, not visible, and don't have a name. The process contains a "soup" of practices.

To easily identify, design, and deploy new practices, we need to let practices become first-class citizens. Process is the result of composing the practices you have selected. To make this possible, we need to be able to separate different practices from one another during design, deployment, and improvement.

In this regard, EssUP is very different from other processes or methods in how it is presented. It embodies an idea new to the process industry—Separation of Concerns (SOC), as in aspect-oriented thinking. When we apply this idea to process development, we generate a fundamentally different process—one that makes it easier and more intuitive to select your tailor-made software process. Most importantly, it will be more natural and intuitive to plan and execute a software project. To illustrate, here are a number of situations where we have applied the idea of SOC:

- Each practice is kept separately from the other practices. You choose only the practices you need without having to deselect activities and artifacts. Just select the practices you want and compose them with one another and with your existing process.
- You can easily separate elements from your existing process and from the elements of the EssUP. This lets you improve what you already have, one practice at a time, in an evolutionary way.

  Starting from a generic platform, you describe your own process using a very simple technique inspired by the game industry. Based on this, you can add practices without requiring a revolutionary adoption of all practices. You take what you need and what you think your organization can adopt without severe risks.

- EssUP separates two different views of the process—the process authors' and software developers' view. In the past, processes have almost entirely focused on the authors' needs. EssUP prioritizes the developers' perspective. It uses techniques from the game industry to develop, teach, apply, and use the process to make it lightweight and agile. And, we promise, much more fun.
- We separate the essentials from the non-essentials. This lets us create a core lightweight process with unprecedented growth potential (hundreds of practices).

We have learned over the years that few people really read process materials, whether in a book or on the Web. So instead of giving developers lots of ignored information, we provide the real essence and let them learn the rest however they want. Some may want to learn it by studying, and there are a lot of articles and books available to do that. Others learn by working with people who have already gained the knowledge. The effect of this separation is that the process is lightweight, and easy to adopt and change.

- It separates explicit knowledge from tacit knowledge in a balanced way. Tacit knowledge is knowledge you acquired one way or the other—you have it in your head. Explicit knowledge exists in the form of descriptions made available to you.

  In the past, processes have tried to capture all related knowledge in an explicit form. Though this is a good ambition, it makes the process heavy. EssUP makes only the essentials explicit. The rest is either tacit or referenced from the essentials. However, the most elegant form of dealing with knowledge is to deliver it through intelligent agents when needed—this is what we call a "smart practice." Such smart practices are now available in Waypointer from Jaczone (www.jaczone.com).

- It is prepared to separate creative work from no-brain work. This lets you spend your time on what humans are good at—being creative—and leaves the no-brain work to intelligent agents. We used the term "prepared" because EssUP doesn't come with agents; agents are add-ons.
- It separates two kinds of artifacts—alphas and betas. We have not yet given them any names. We think the distinction between alphas and betas are one of the most fundamental in all projects. Tentatively you may replace alpha with "key thing" and beta with "evidence" (of a key thing).

  Alphas are the most important things software projects have, whether they exist in a described form or not. Actually, the alphas are not specific for any particular process. For instance, every software project has these alphas: project, implemented system, backlog, risk. Each alpha has a set of betas: A project alpha may have a project plan, an iteration plan. A risk may have a risk list. A backlog may have a feature list and change list.

  Betas are evidence of alphas. They can be descriptions, diagrams, flow charts, or whatever you like to document or not document. "Not document" means that teams keep the knowledge in their heads.

  Whereas alphas are stable and process independent, betas may be different based on what practices you have chosen to use. The separation of the alphas from the betas lets you keep the project documentation at a minimum. You can in a precise way discuss how much to document. This lets you be agile in a disciplined way. You can separate the essentials from the less essential.

The Basis Of the Process

At the heart of EssUP are a number of simple and proven practices that can be used as the foundation for all styles and scale of software development. These practices have all been designed so that they can be adopted individually or in any combination you desire. This makes the process easy to adopt and enables it to provide a foundation for the creation and assembly of a process that truly meets your needs, the needs of developers, and the needs of your development organization.

EssUP comes with five foundation practices and three work practices to ease the introduction of the process into your organization. The five foundation practices address technical development work. To complement the technical foundation provided by the development practices, three other practices promote effective team work and process improvement.

Test is everywhere. We believe in the saying "Whatever you do, you are not done until you have verified that you did what you wanted to do," or "Everyone is his own cleaner." We make test an integral part of everything we do. Use-Case Essentials include test-driven design because use cases are early test cases. Component Essentials include unit testing of components.

This set of practices provides you with the foundation to take an agile approach to process planning and implementation. You can adopt all the practices, just the practices you need, an individual practice, or even a partial practice. You can mix and match the practices to meet your needs, write your own practices to extend the process, and mix in your own existing practices to build on your own experiences. This is a significant difference to traditional processes, which are developed with all their practices tightly intertwined.

Presented In a Radical New Way

The changes to the process go beyond the separation of concerns, presenting the process material itself in a radically new way.

Each practice is presented as a set of process cards that contain the elements you need to build your process, including competencies, activities, and artifacts. The cards help you build and use the process. The card metaphor makes the process itself agile and easy to use. Displayed electronically or presented as physical cards, they are easily manipulated to facilitate process adoption, project planning, and to provide handy reference for practitioners. The cards bring the process to life and make it more visible than a web site or book.

Figure 1 presents samples from the Use-Case Essentials practice. Figure 1(a) is an artifact card, actually a beta, 1(b) an activity card, and 1(c) a competency card. For each kind of card there is a 2-4 page guideline (Figure 2) presenting the most essential information needed to put the cards into practice. They also link into a broader set of references, scripts, tools, templates, and samples. For example, the activity guidelines include an introduction, participant information, completion criteria, and a set of hints and tips with a list of common mistakes to avoid. This information forms the essential information to give you practical advice.

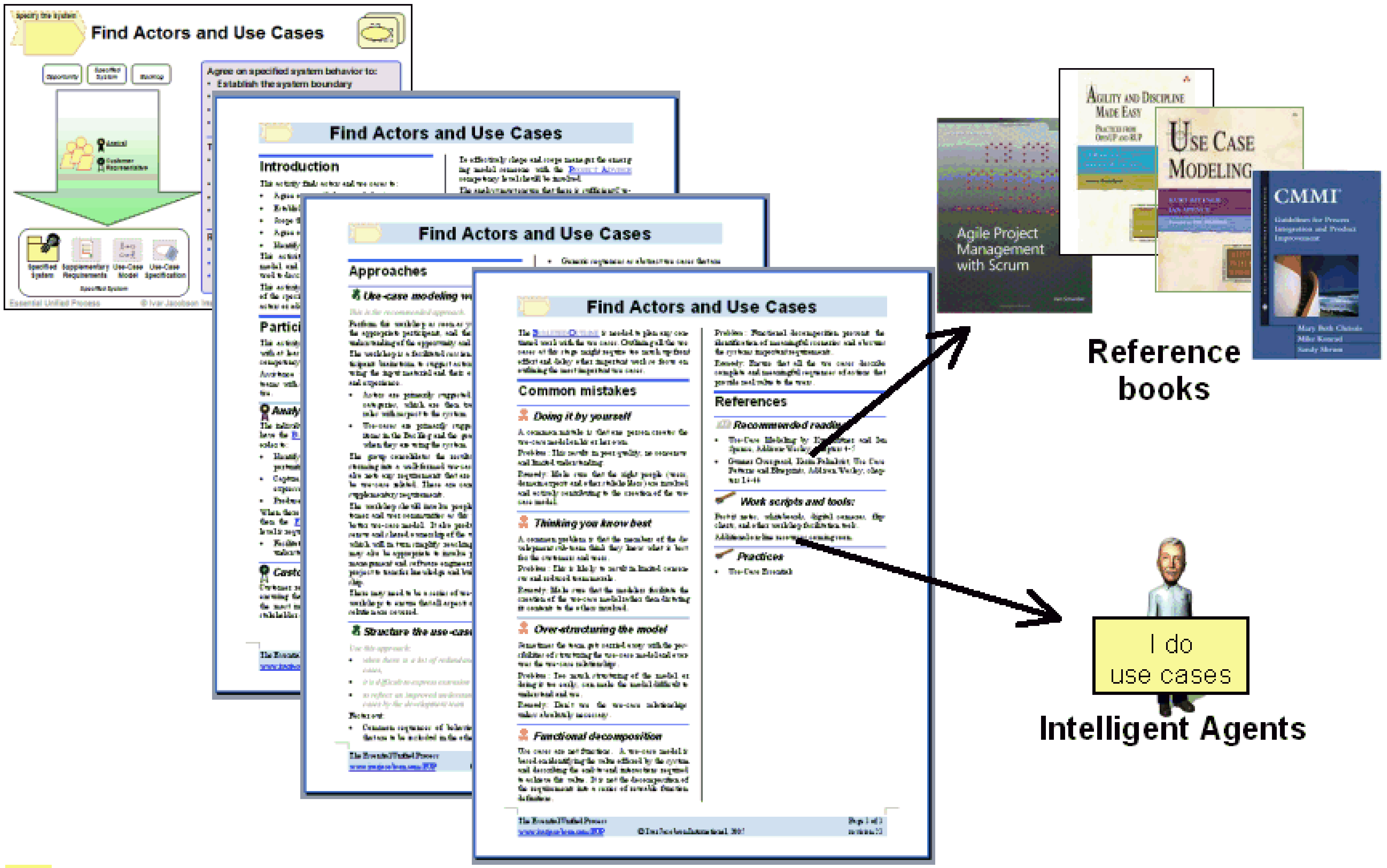
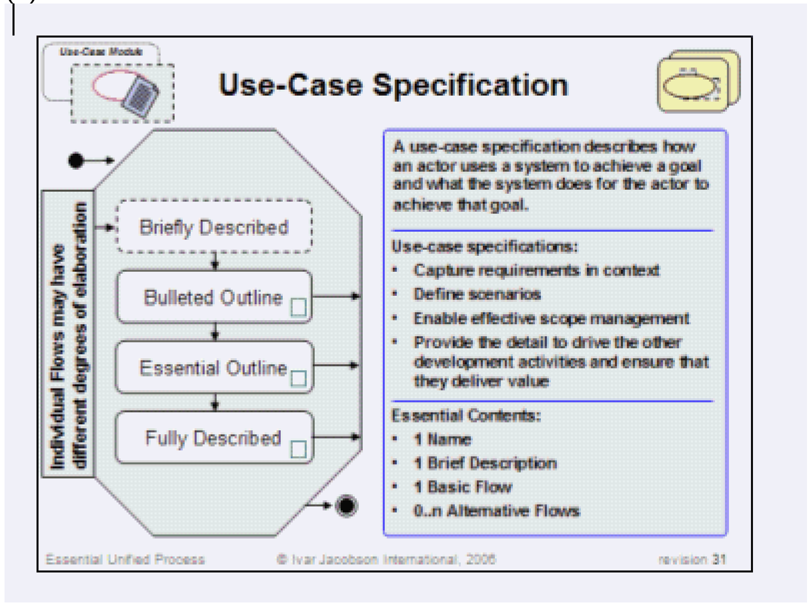

Find Actors and Use Cases
Introduction
Approaches
Use-case modeling workshop
Structure the use-case model
Common mistakes
Doing it by yourself
Thinking you know best
Over-structuring the model
Functional decomposition
References
Recommended reading
Work scripts and tools:
Practices
Agile Project Management with Scrum
Agility and Discipline Made Easy
Use Case Modeling
CMMI
Reference books
I do use cases
Intelligent Agents

Figure 2: Guidelines.

(a)


Use-Case Module
Individual Flows may have different degrees of elaboration
Briefly Described
Bulleted Outline
Essential Outline
Fully Described

Use-Case Specification

A use-case specification describes how an actor uses a system to achieve a goal and what the system does for the actor to achieve that goal.

Use-case specifications:

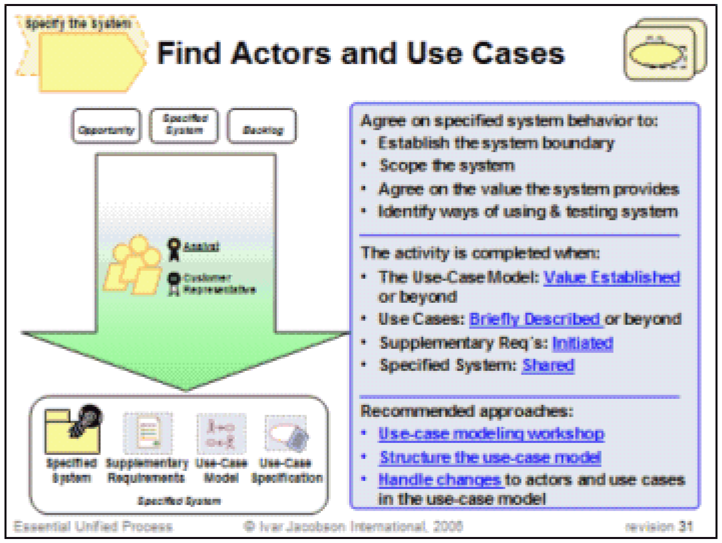
- Capture requirements in context
- Define scenarios
- Enable effective scope management
- Provide the detail to drive the other development activities and ensure that they deliver value

Essential Contents:

- 1 Name
- 1 Brief Description
- 1 Basic Flow
- 0..n Alternative Flows

Essential Unified Process © Ivar Jacobson International, 2006 revision 31

(b)



Specify the System
Opportunity
Specified System
Backlog
Analyst
Customer Representative
Specified System
Supplementary Requirements
Use-Case Model
Use-Case Specification
Specified System

Find Actors and Use Cases

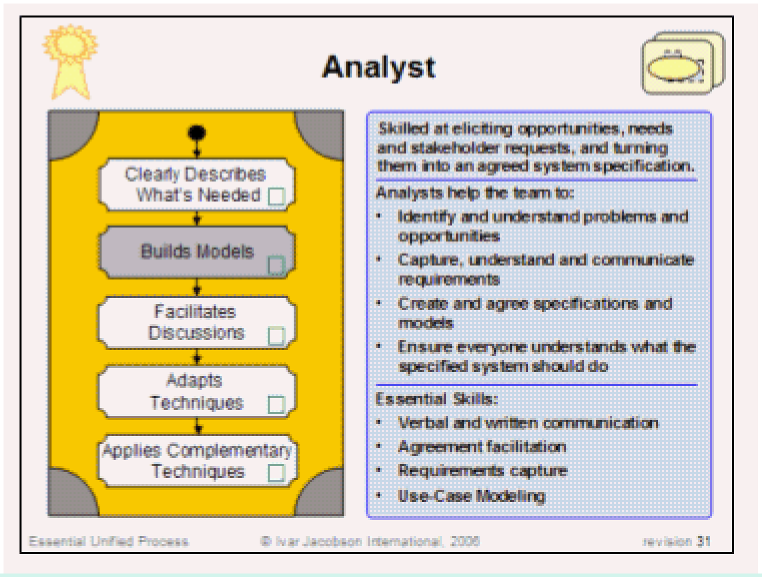
Agree on specified system behavior to:
- Establish the system boundary
- Scope the system
- Agree on the value the system provides
- Identify ways of using & testing system

The activity is completed when:
- The Use-Case Model: Value Established or beyond
- Use Cases: Briefly Described or beyond
- Supplementary Req's: Initiated
- Specified System: Shared

Recommended approaches:
- Use-case modeling workshop
- Structure the use-case model
- Handle changes to actors and use cases in the use-case model

Essential Unified Process © Ivar Jacobson International, 2006 revision 31

(c)


Clearly Describes What's Needed
Builds Models
Facilitates Discussions
Adapts Techniques
Applies Complementary Techniques

Analyst

Skilled at eliciting opportunities, needs and stakeholder requests, and turning them into an agreed system specification.

Analysts help the team to:

- Identify and understand problems and opportunities
- Capture, understand and communicate requirements
- Create and agree specifications and models
- Ensure everyone understands what the specified system should do

Essential Skills:

- Verbal and written communication
- Agreement facilitation
- Requirements capture
- Use-Case Modeling

Essential Unified Process © Ivar Jacobson International, 2006 revision 31

Figure 1: (a) Artifact card; (b) activity card; (c) competency card.

How Is the Process Packaged?

The process is supplied with the base set of foundation practices delivered in card form with supporting guidelines. The foundation practices are designed to be extended with booster packs. These can be written by yourself to meet your specific need and they will be supplied by others. They include packs for practices, such as Service-Oriented Architecture, Business Modeling, Enterprise Architecture, Pair Programming, CMMI, and the like.

The cards can be applied in a number of different ways. For instance, they can be used to:

- Build the deck you need to support your project.
- Combine the cards to create work items for team members or new process elements.
- Have many card instances to represent the actual deliverables and tasks of the project.
- Annotate the cards to add the information that is relevant to your own project.
- Capture instance data that can be applied to your implementation of the cards.
- Deal the cards to the project team, to provide them with the process information that they need.
- Draw a card as a project team member so that you will have the information relevant to you.

- Swap cards within your team.
- Write new cards to provide support for your own environment.

Through the three Team Working Practices (Table 2) covering product essentials, process essentials, and team essentials, the process guides you in an agile way through its use and implementation in team environments. This can be done with a physical set of cards or by using an electronic representation of them. The process helps promote good working practices within a team environment. This is closely supported by the Visual Studio Team System environment, in which the process will be integrated. The process will also be made available on top of the Eclipse Process Framework.

The process is provided with a number of game boards that provide a holding place for your cards. A set of instructions is also provided for straightforward implementation of the process. This can be a physical board to go with the physical cards or can be represented in electronic form.

How Do You Implement the Process?

You implement the process by improving what you already have, one practice at a time, in an evolutionary way. You take what you need and what you think your organization can adopt, without taking any severe risks. You deal out the cards to the project team, giving them the information that they need to focus upon. Cards contain the essential information, with added project-specific instructions that the project manager can add. In the past, processes have almost entirely focused on the authors' needs. EssUP prioritizes the developers' perspective. Its techniques are used to develop, teach, apply, and use the process to make it lightweight and agile.

Final Words

The Essential Unified Process:

- Concentrates on the essentials applicable to all your projects.
- Enables you to build on the skills that you already possess.
- Provides guidance on implementing a consistent approach.
- Focuses on enhancing the skills of the people involved in development.
- Adds just enough process to address your project risks.

Our goal with the EssUP is a long-term vision: To move from a "process" era when everyone is forced to think about process to an "invisible process" era when we don't talk specifically about process, but assume it as a given.