ESSENSE – A Kernel of Essentials for Software Engineering

Brian Elvesæter, SINTEF ICT

Method Engineering – ME’11

21 April 2011, Paris, France

Outline

- Request for Proposal (RFP)
  - Problem statement
  - Desired solution
  - Objectives of the RFP
  - RFP development history and team
  - Method architecture and the Kernel approach
  - Requirements: The Kernel
  - Requirements: The Language
  - Proposed RFP schedule
- Questions

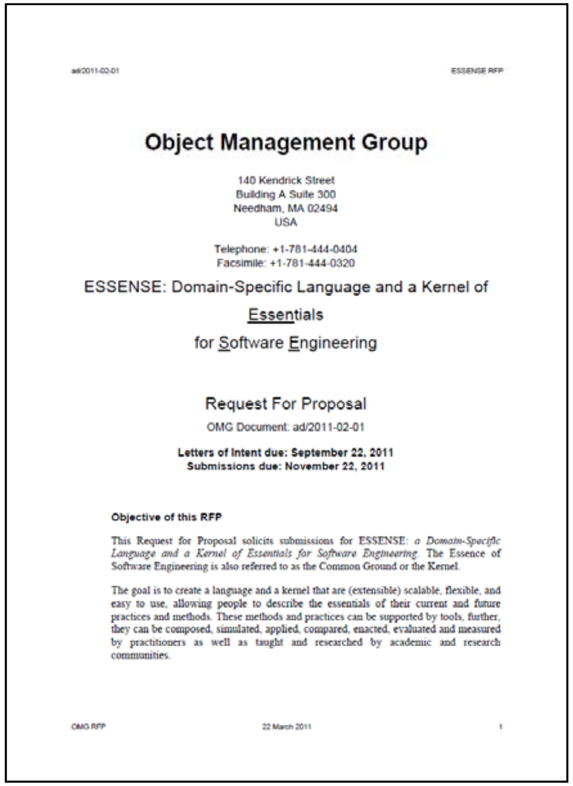
Problem statement

- Traditional software engineering and process engineering approaches are
  - viewed by development teams as being
  - too heavyweight and inflexible.
- Software processes defined by separate process engineers typically
  - do not leave enough flexibility for a development team to customize and tailor the process they use,
  - not just at the beginning, but continuously as necessary over the course of a development effort.
- As a result, the use of many good practices and processes
  - is often missing or ad hoc in many development efforts,
  - limiting the ability of development teams to be effective and scalable
  - while remaining flexible and agile.

Desired solution

- Agile software development is often effectively supported by the use of significant software frameworks that provide:
  - a toolkit of components (libraries and templates)
  - an easy-to-use scripting language for flexibly composing the components.
- A similar kind of framework is desired for the flexible creation of software development methods.
  - a set of practices out of which methods can be composed
- To allow the broadest possible applicability, what needs to be standardized are not the practices themselves, but the common ground of underlying concepts and principles used to define various practices.
  - an easy-to-use practitioner-oriented modeling language (DSL) is needed to define practices based on the common ground and composing them into methods.
- The goal is to support a development team, i.e. the practitioners
  - in defining, refining and customizing themselves the process they are actually using during the course of a software development effort.

ESSENSE RFP Objectives



ad/2011-02-01

ESSENSE RFP

Object Management Group

140 Kendrick Street
Building A Suite 300
Needham, MA 02494
USA

Telephone: +1-781-444-0404
Facsimile: +1-781-444-0320

ESSENSE: Domain-Specific Language and a Kernel of Essentials for Software Engineering

Request For Proposal

OMG Document: ad/2011-02-01

Letters of Intent due: September 22, 2011
Submissions due: November 22, 2011

Objective of this RFP

This Request for Proposal solicits submissions for ESSENSE: *a Domain-Specific Language and a Kernel of Essentials for Software Engineering*. The Essence of Software Engineering is also referred to as the Common Ground or the Kernel.

The goal is to create a language and a kernel that are (extensible) scalable, flexible, and easy to use, allowing people to describe the essentials of their current and future practices and methods. These methods and practices can be supported by tools, further, they can be composed, simulated, applied, compared, enacted, evaluated and measured by practitioners as well as taught and researched by academic and research communities.

OMG RFP 22 March 2011 1

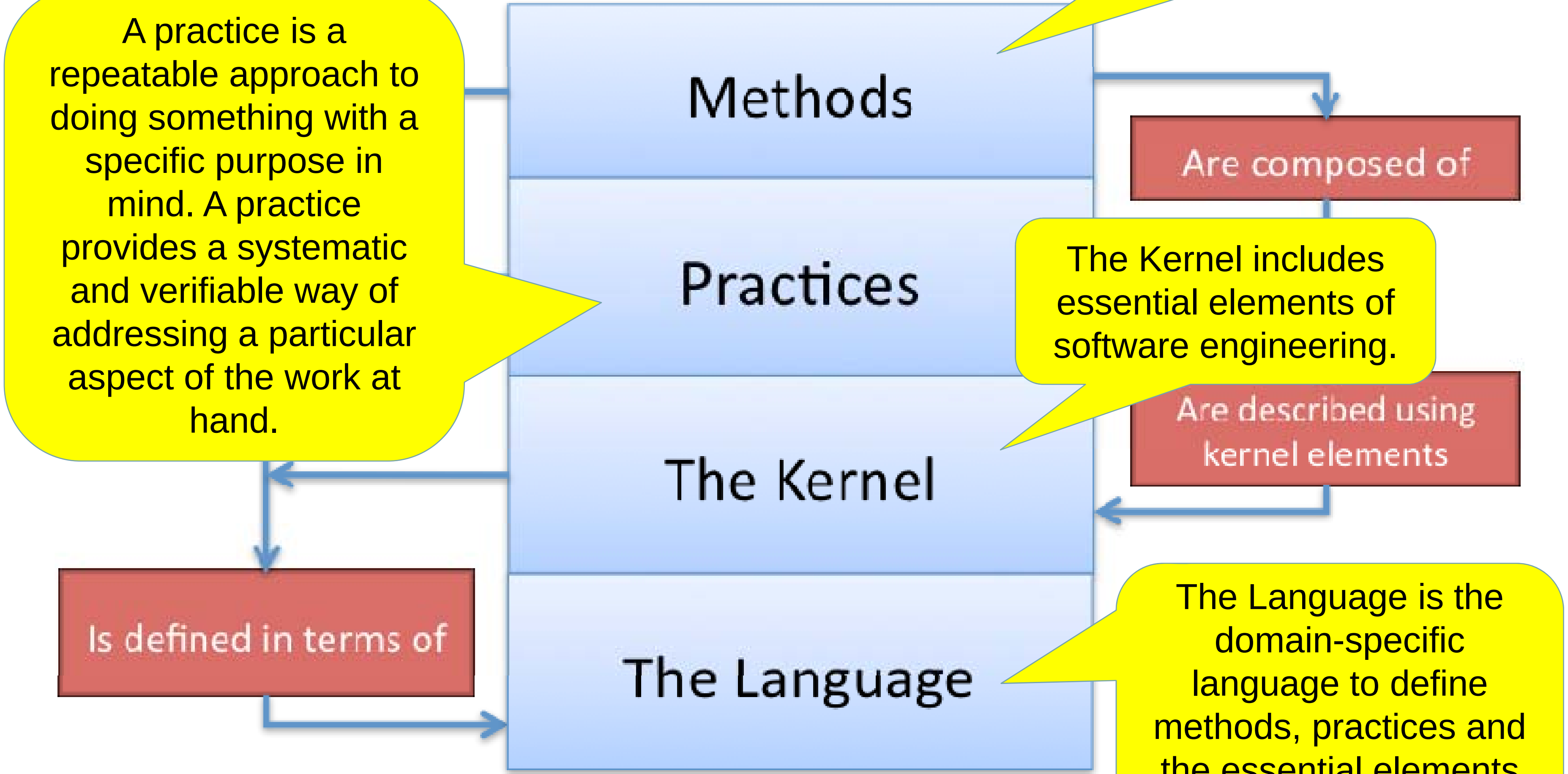
- Solicits submissions for ESSENSE
  - a Domain-Specific Language and
  - a Kernel of Essentials for Software Engineering (aka. the Common Ground)
- **Goal:** a language and a kernel that are
  - scalable, extensible, and easy to use,
  - allowing people to describe the essentials of their current and future methods and practices.
- **Tool support:** enable methods and practices to be
  - composed, compared, evaluated,
  - tailored, used, adapted, simulated and measured
  - by practitioners as well as taught and researched by academic and research communities.

RFP development history & team

- Version 0:
  - December 2010, Issue list, OMG technical meeting, Santa Clara, USA
- Version 1a:
  - February 2011, Draft RFP
    - http://www.omg.org/cgi-bin/doc?ad/2011-02-01
- Version 1b:
  - March 2011, Updated RFP, AB review, OMG technical meeting, Washington DC, USA
    - http://www.omg.org/cgi-bin/doc?ad/2011-03-01
- Version 2a/b:
  - May/June 2011, Revised RFP, OMG technical meeting, Salt Lake City, USA

- Core team
  - Arne-Jørgen Berre (SINTEF)
  - Dave Cuningham (Fujitsu UK)
  - Brian Elvesæter (SINTEF)
  - Shihong Huang (FAU)
  - Ivar Jacobson (IJI)
  - Paul McMahon (PEM Systems)
  - Ed Seidewitz (Model Driven Solutions)
  - Ed Seymour (Fujitsu UK)
- Other contributors
  - SEMAT
  - Adaptive Systems
  - Cordys
  - SOFTEAM
  - ESI-Tecnalia
  - ...

Method architecture



A method is a composition of practices. Methods are enactable.

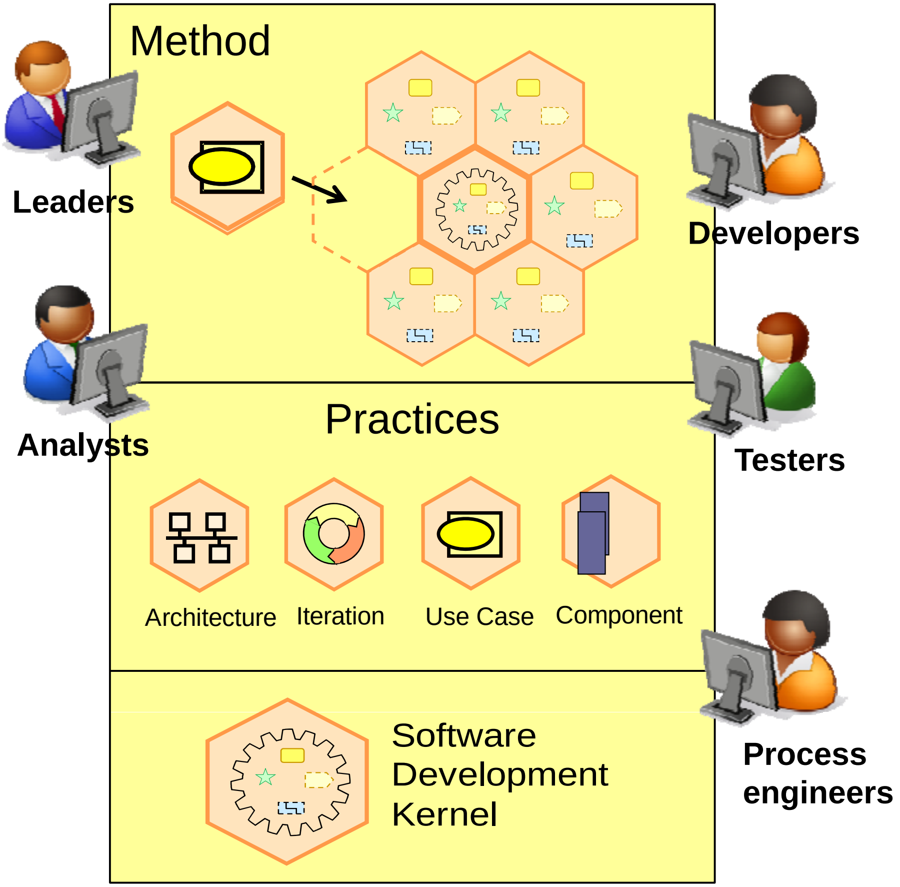
A practice is a repeatable approach to doing something with a specific purpose in mind. A practice provides a systematic and verifiable way of addressing a particular aspect of the work at hand.

The Kernel includes essential elements of software engineering.

The Language is the domain-specific language to define methods, practices and the essential elements of the Kernel.

A key idea is the existence of a Kernel

- The Kernel is very small, extracted from a large number of methods
- It contains a starting point (slots) for the things that every process has, e.g.
  - work, team, requirement, software system, opportunity and stakeholder
- The Kernel is practice and method agnostic.

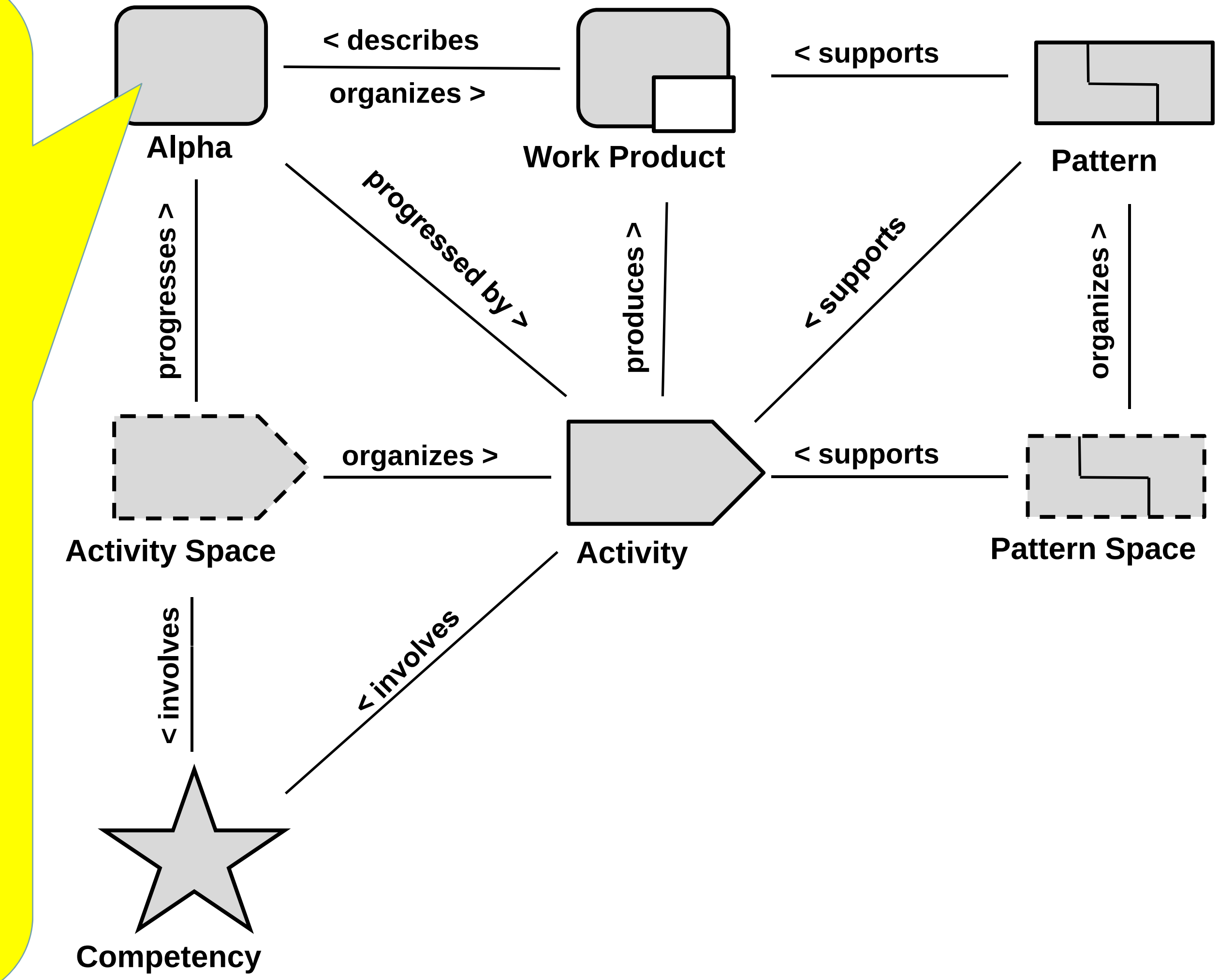


Kernel: Straw man (illustrative conceptual model)

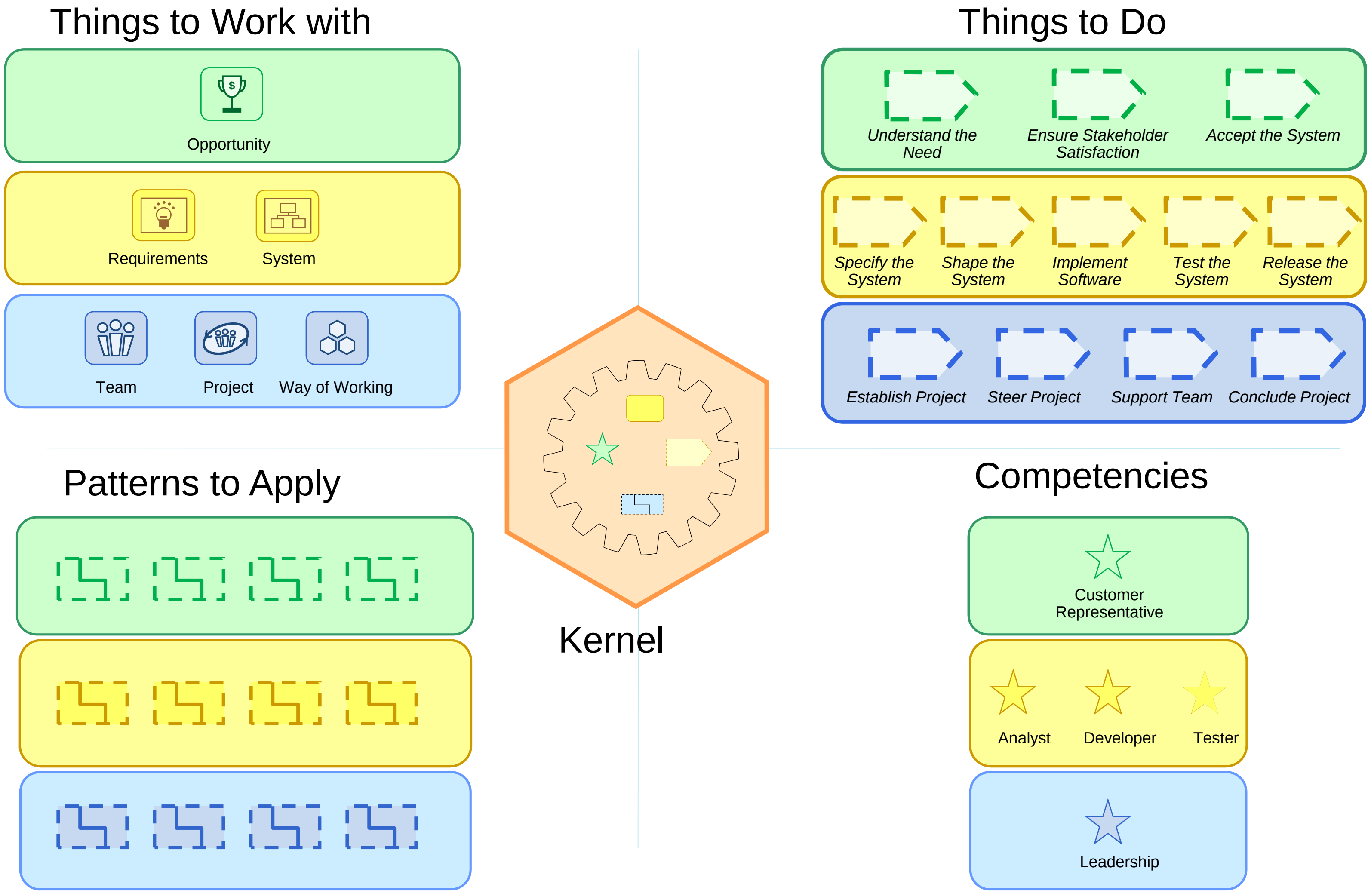
Alphas (Abstract-Level Progress Health Attributes) represent things to work with that subsume and encapsulate work products at a higher level of abstraction.

(1) relevant to an assessment of the project's progress (stated objectives such as deadlines, costs, quality)

(2) can be determined (directly or indirectly) in terms of the current state of the project's work products



Kernel: Straw man (illustrative concrete syntax)



Things to Work with

Opportunity

Requirements System

Team Project Way of Working

Things to Do

Understand the Need

Ensure Stakeholder Satisfaction

Accept the System

Specify the System

Shape the System

Implement Software

Test the System

Release the System

Establish Project

Steer Project

Support Team

Conclude Project

Patterns to Apply

Kernel

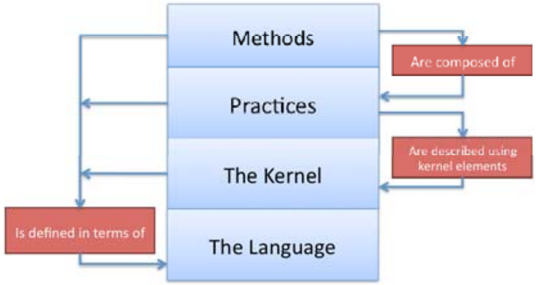
Competencies

Customer Representative

Analyst Developer Tester

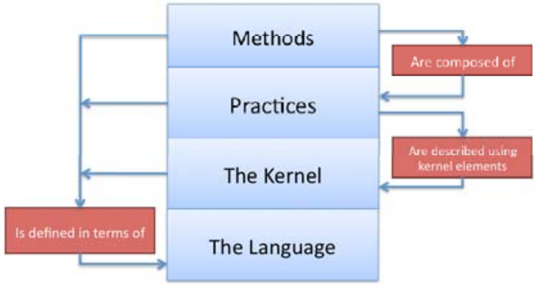
Leadership

Requirements: The Kernel



- Definition
  - Expressed in the Language.
  - Encompass the fundamental concepts in the Software Engineering domain, including definitions of elements and their significant relationships.
- Conciseness
  - Only include a small set of elements that are truly essential.
  - The number of essential elements is expected to be closer to 10 than to 100.
- Scope
  - Cover from the smallest projects to large systems and systems-of-systems.
- Broad practice coverage
  - Support many different practices used by significant segments of the industry.
- Broad lifecycle coverage
  - Accommodate various lifecycle models used by significant segments of the industry.
- Broad technology coverage
  - Adaptable to a wide range of software technologies (programming languages, specification languages, graphical notations, software tools) used by significant segments of the industry.
- Comparison
  - Provide a basis for the comparison of methods and practices to see which are suitable for a given situation.
- Measurement
  - Provide a basis for the measurement of methods and practices, both to enable performance evaluation and to guide evaluation and validation in research.
- Extension
  - Ability to add practices, levels of detail and lifecycle models.

Requirements: The Language



- Definition
  - Abstract syntax model defined in MOF (Meta-Object Facility).
  - Formal static and operational semantics defined in terms of the abstract syntax.
  - Graphical concrete syntax that formally maps to the abstract syntax.
  - Textual concrete syntax that formally maps to the abstract syntax.
- Description
  - Support the description of practices and methods in terms of the essential elements of the Kernel.
- Composition
  - Support the composition of practices to describe existing and new methods.
- Work Progress
  - Allow the representation of work progress.
  - (For example, describing a practice that involves iterative development requires describing the starting and ending states of every iteration.)
- Enactment
  - Support the enactment of methods, both as used to help plan endeavors and as applied (or executed) as part of the day-to-day activities in real projects.

Proposed RFP schedule

Event or Activity	Actual Date
Preparation of RFP by TF	ADTF – December 8, 2010
RFP placed on OMG document server	February 21, 2011
Discussion of RFP by Architecture Board Review by TC	March 23, 2011
Discussion in ADTF on the RFP	March 24, 2011
Approval of RFP by Architecture Board Review by TC	June, 2011
TC votes to issue RFP	June, 2011
LOI to submit to RFP due	September, 2011
Initial Submissions due and placed on OMG document server (“Four week rule”)	November, 2011
Voter registration closes	November, 2011
Initial Submission presentations	December, 2011
Preliminary evaluation by TF	December, 2011
Revised Submissions due and placed on OMG document server (“Four week rule”)	May, 2012
Revised Submission presentations	June, 2012
Final evaluation and selection by TF Recommendation to AB and TC	September, 2012
Approval by Architecture Board Review by TC	September, 2012
TC votes to recommend specification	September 2012
BoD votes to adopt specification	September, 2012

Questions

- Email:
  - brian.elvesater@sintef.no
- OMG website:
  - http://www.omg.org
- SEMAT website:
  - http://www.semat.org