

Webinar: A Pluggable Process Architecture

April 14th, 2005

Brian G. Lyons
Number Six Software, Inc.

Sammy Wahab
Osellus Inc.

A Pluggable Process Architecture

Brian G. Lyons, Number Six Software, Inc.

- **A Pluggable Process Architecture (PPA) is**
 - **A framework for process components to be assembled into a software engineering process**
 - **Resulting process will be consistent**
 - **Resulting process will be complete**
 - **Process components will be reusable**

- **Framework defines**
 - **A categorization of process components**
 - **An organized set of packages containing said components**
 - **A set of rules defining a well-formed process**

- **A Pluggable Process Architecture is valuable to**
 - **Organizations that must define different processes for different project types**
 - Development organizations
 - Consulting organizations
 - **Organizations that must define an overarching process for suppliers to work within**
 - Large development clients (e.g. government agencies)
 - Prime contractors

For Whom (cont.)

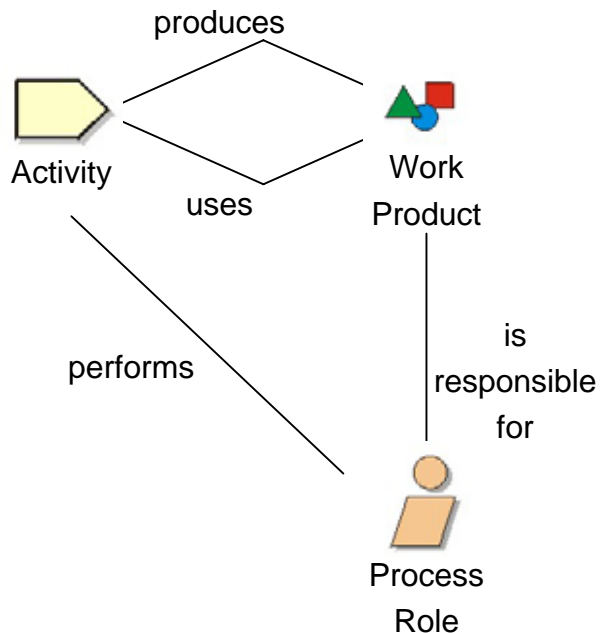
- **A Pluggable Process Architecture is valuable to**
 - **Organizations targeting CMMI-compliance**
 - **One or more life-cycle model descriptions are described**
 - **Pluggable elements repository makes up the process asset library**
 - **Organization's Set of Standard Processes (OSSP)**
 - **Appropriate integration among process pieces is ensured**
 - **Institutionalized process is tailored in a mature fashion**
 - **Tailoring Guidelines**
 - **Subcontractor adherence supports supplier agreement management**

- **There are many development processes out there and each has:**
 - **A notion of a lifecycle with phases and milestones**
 - **Coverage of key disciplines**
 - **(i.e. Requirements, Analysis & Design, Implementation, Test, Project Management)**
- **Large organizations with multiple projects will need to apply appropriate methods for each unique project**
- **Disparate processes have disparate methods that cannot be assembled to create a complete process**
- **Mature organizations will want a cohesive set of integratable methods**

- **Every process needs one set of phases**
- **Every process needs a method to support each of a set of core disciplines**
 - **Requirements**
 - **Analysis & Design**
 - **Implementation**
 - **Test**
 - **Project Management**
- **These two notions need a shared perspective on key work products and process roles**
- **All these process components are otherwise independent**

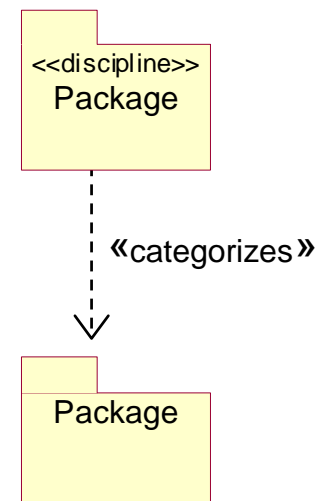
- **The Software Process Engineering Metamodel is a UML extension mechanism used to describe software development processes**

Key Classes



Process Organization

Classes



- **SPEM is a metamodel**
 - **Can be used to describe a software development process**
- **The PPA is a framework**
 - **An overarching organization of a model**
 - **A set of constraints, idioms, and patterns**
 - **Conforms to the metamodel**



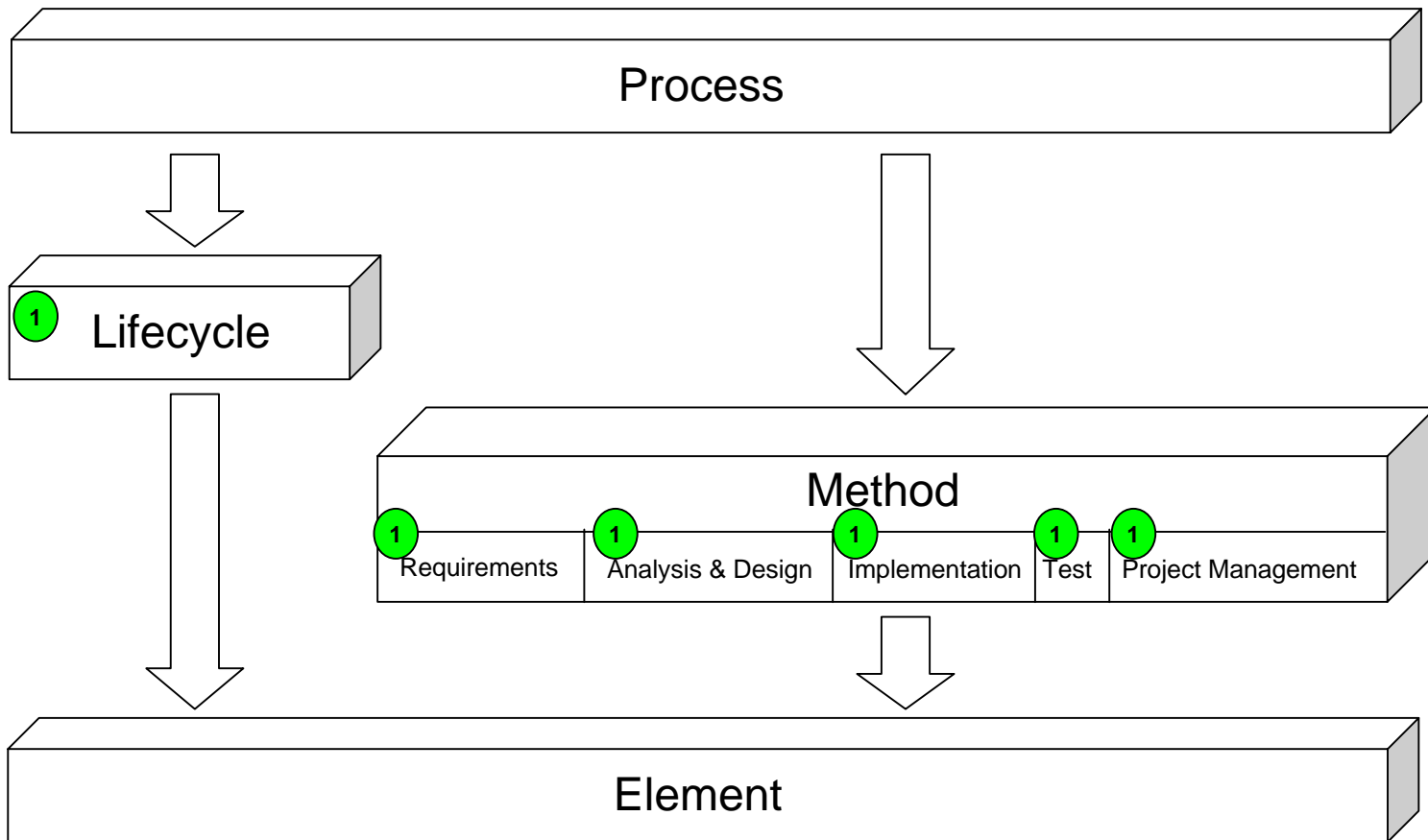
Do We Need a Pluggable Process Architecture?

- **The SPEM is sufficient to define a process**
- **The SPEM does not provide a set of policies to promote interoperable process components**
- **We need something to fill the gap and ensure completeness and consistency**

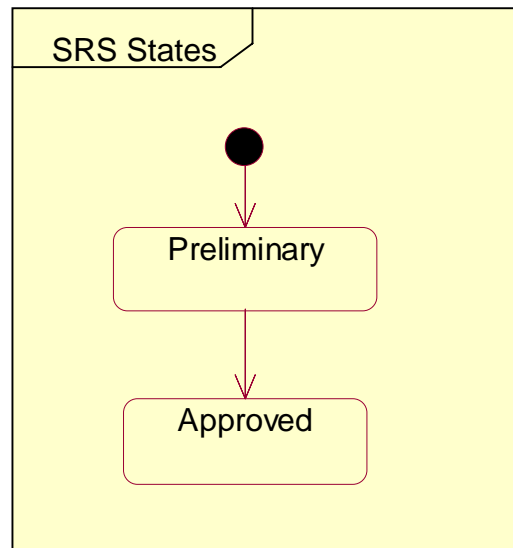
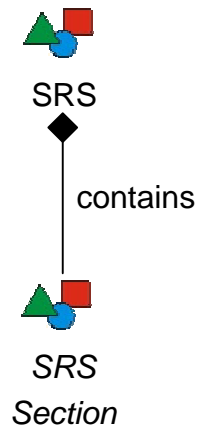
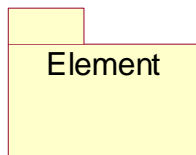
PPA: A Revolutionary Idea?

- **This notion is not new**
- **The SPEM RFP required “Process Patterns and/or Components”**
 - **Plug-and-play patterns**
 - **Support for reusable process engineering components**
- **Yet the SPEM provides little support for this notion**

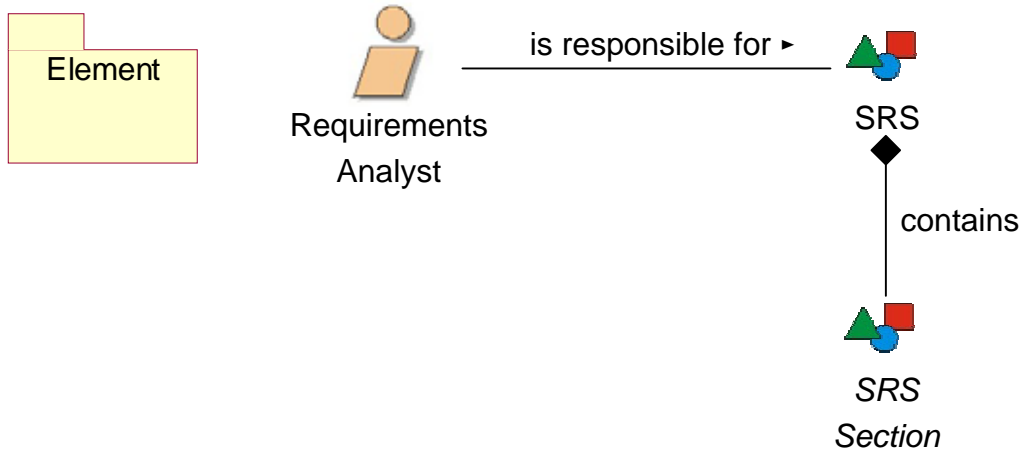
A Layered Process Architecture



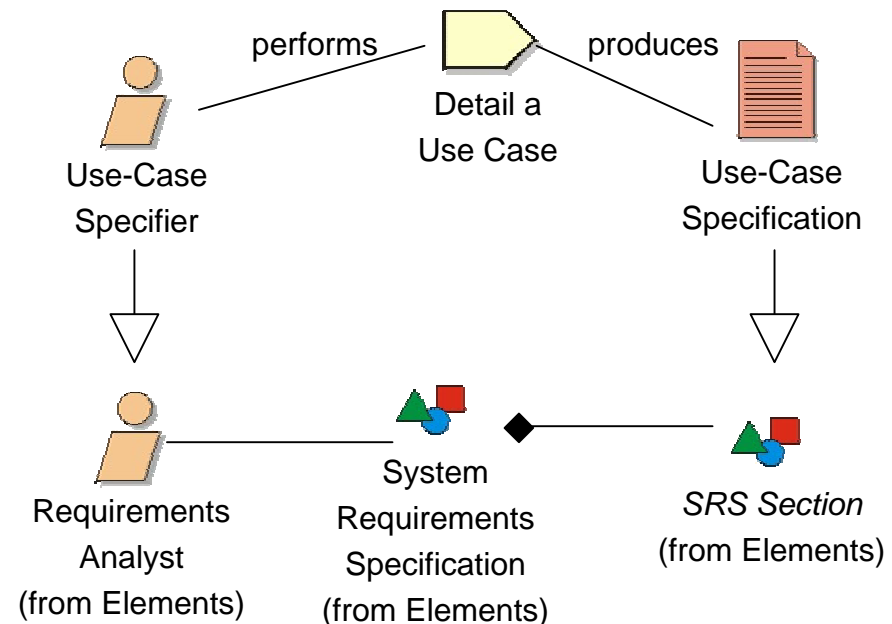
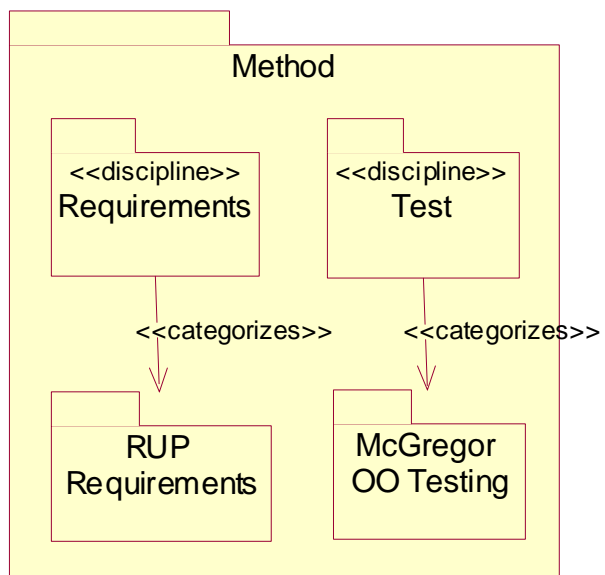
- **Contains reusable work products**
 - **Key work products typically have states**
 - **Key work products that will provide milestone completion criteria**
 - **Often mega-work-products**
 - (e.g. SRS or Design Model)



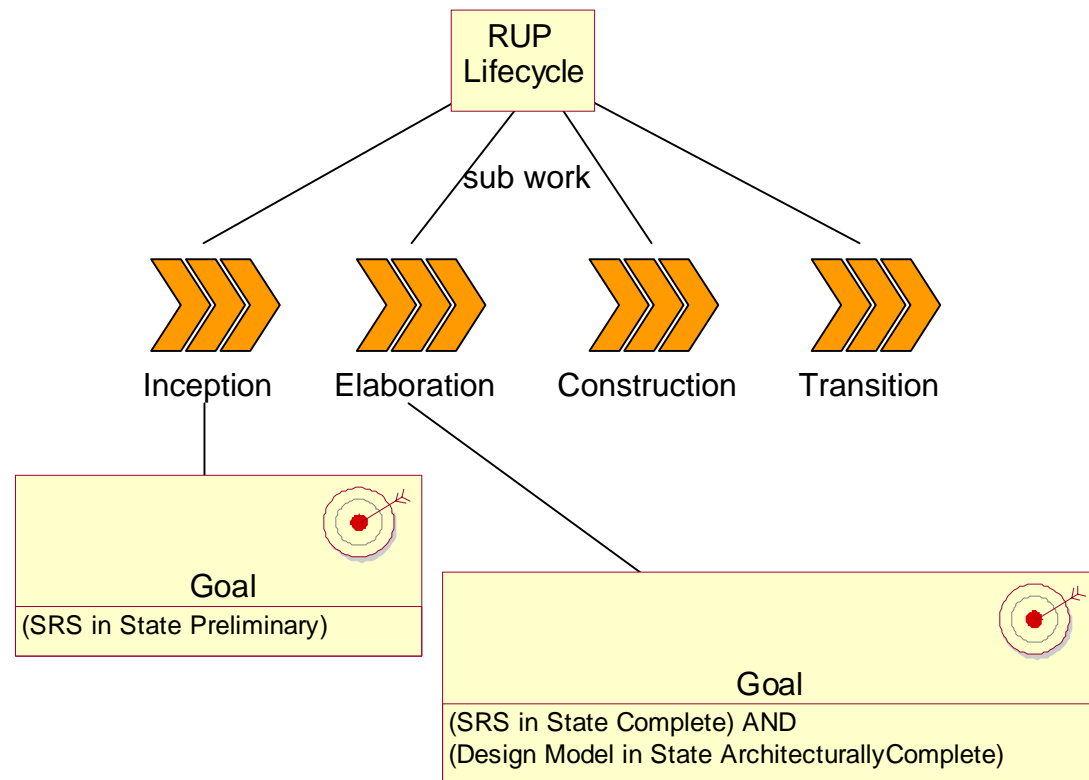
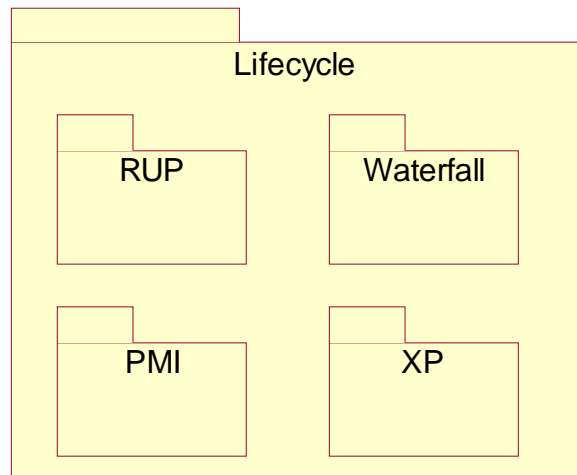
- **Contains reusable roles**
 - **To be responsible for work products**
 - **Often mega-roles**
 - (e.g. System Analyst, Designer)



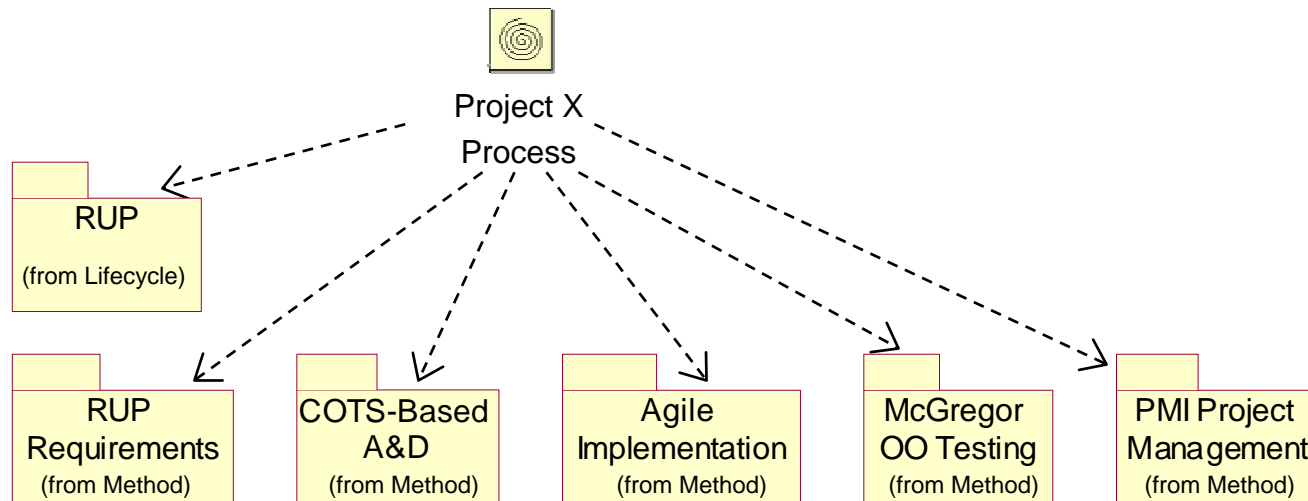
- **Defines disciplines**
- **Contains a number of method packages**
 - Each falls within one discipline
- **Method packages contain sub-process details**



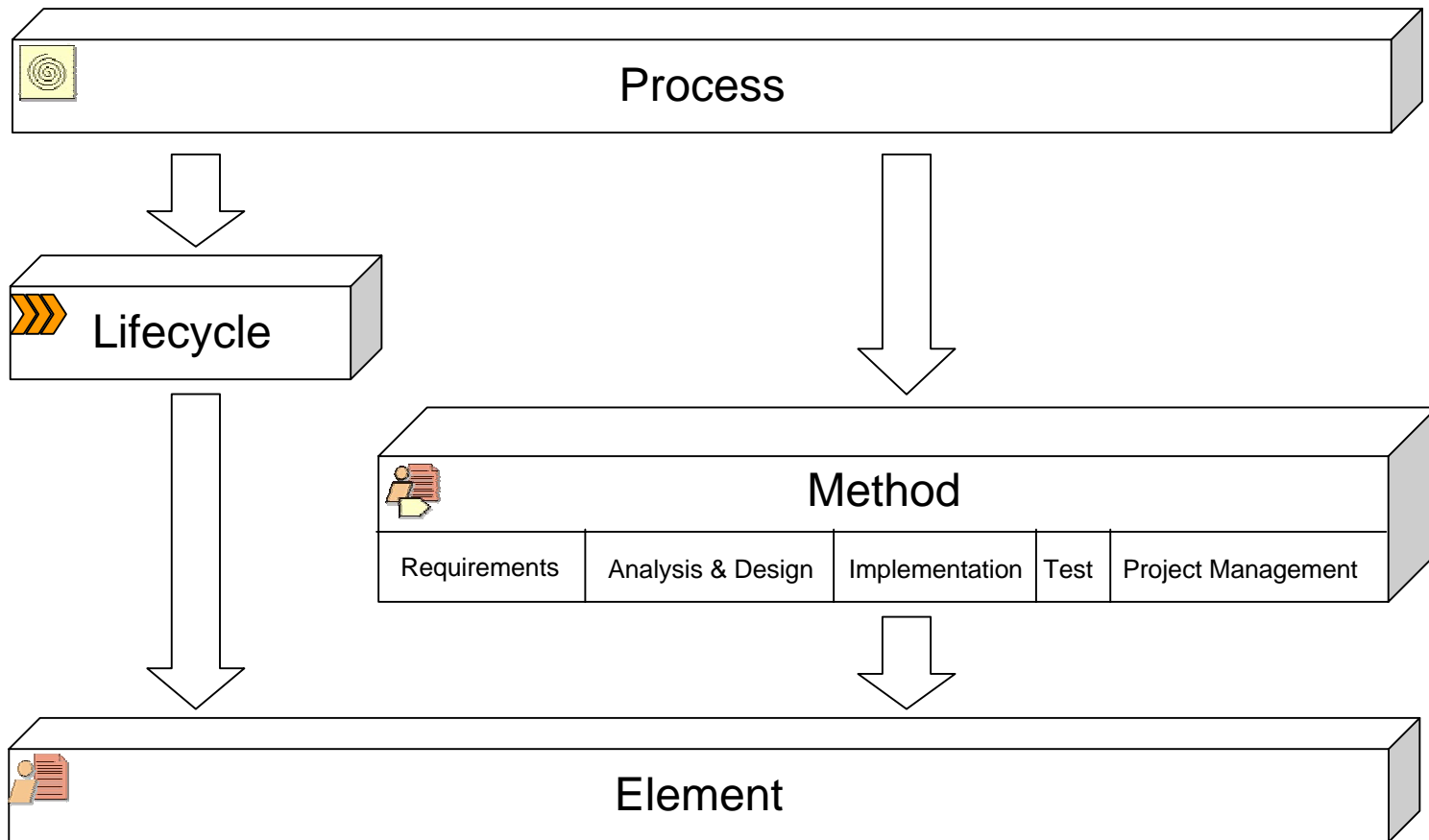
- Includes a number of phase work definitions
- Phases are independent of methods
- Milestones depend on work products



- Contains packages that represent complete processes to be applied
- Distinct rules for being well formed
 - Must import exactly one lifecycle package
 - Must import one method per discipline



A Layered Process Architecture



No Need for “Unification”

- **SPEM assumes disparate ProcessComponents must be “unified”**
- **Inputs and output WorkProduct must be “made identical”**
- **Similar issues with ProcessRoles, Templates, etc.**
- **This sloppy, manual process is no longer needed**

- RUP Lifecycle with MIL-STD-2167A Requirements



Inception

LCO



System
Requirements
Specification
[Preliminary]



Elaboration

LCA



System
Requirements
Specification
[Approved]



Construction



Transition



Specify
System



SSS



Analyze
Requirements



SRS



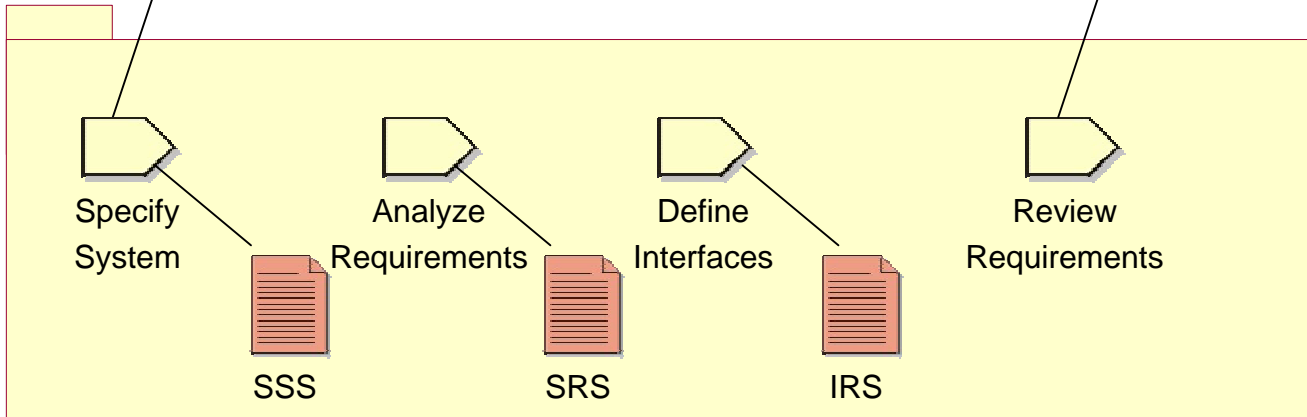
Define
Interfaces



IRS

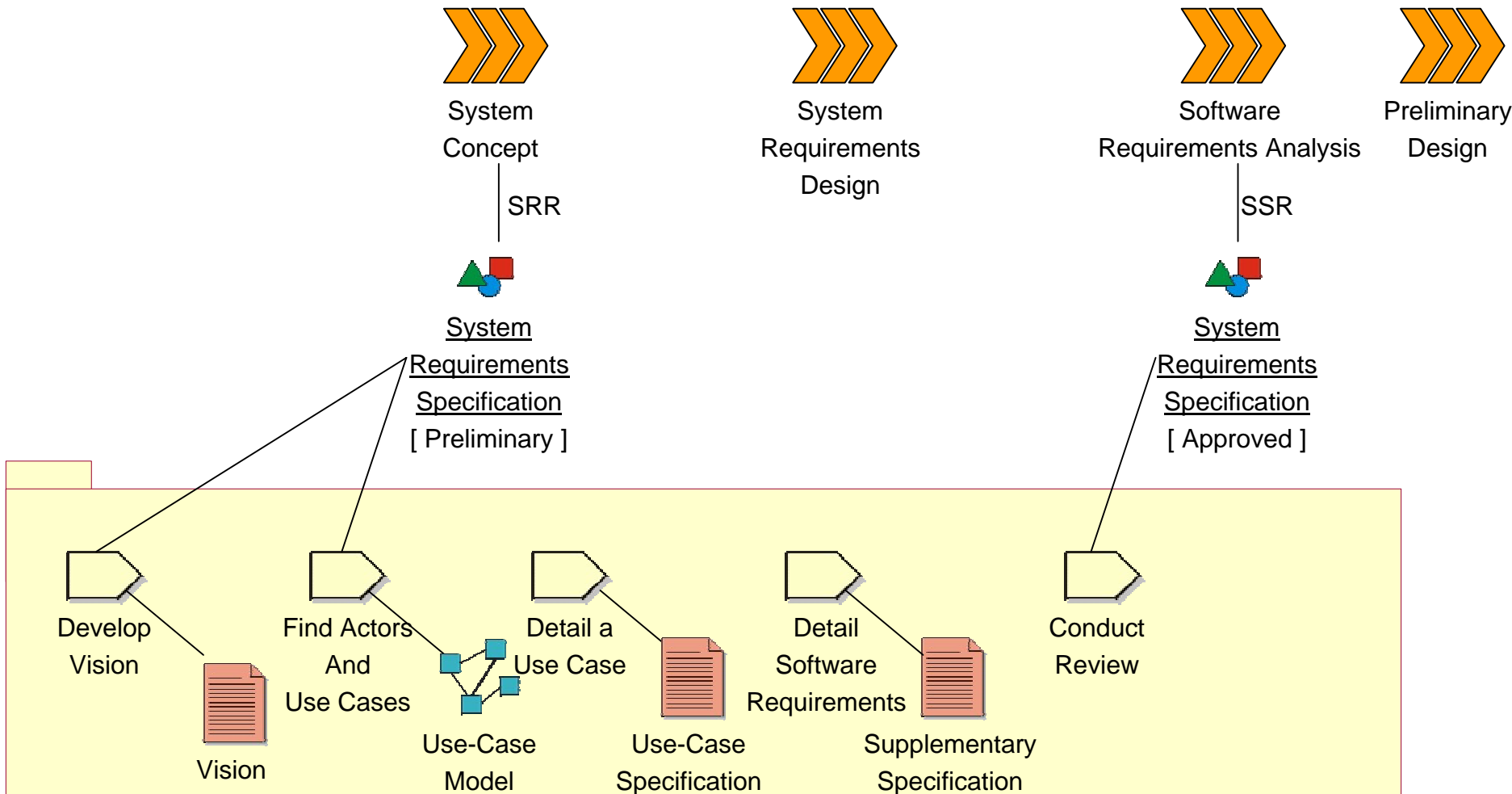


Review
Requirements

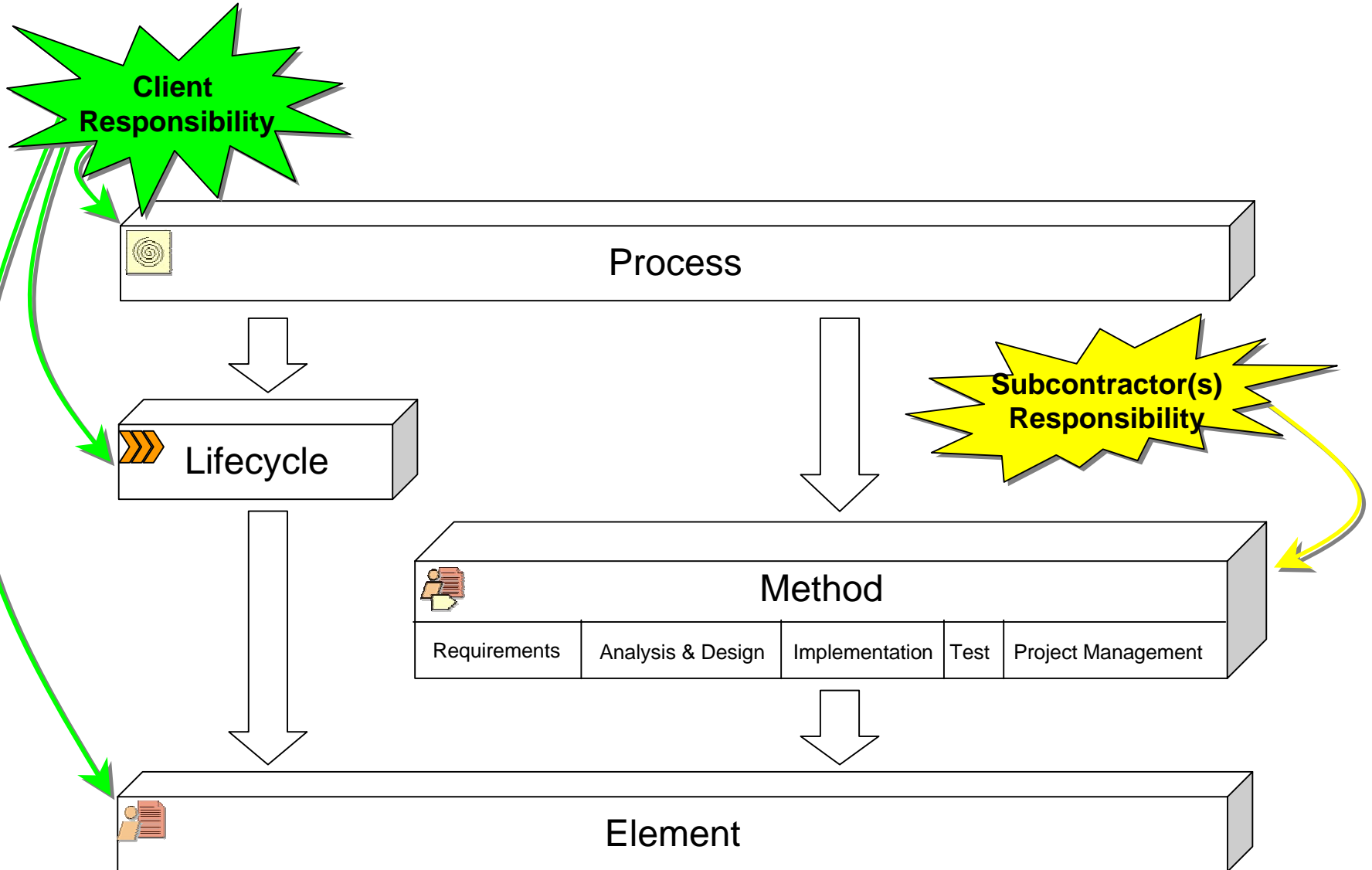


Requirements in Context Example 2

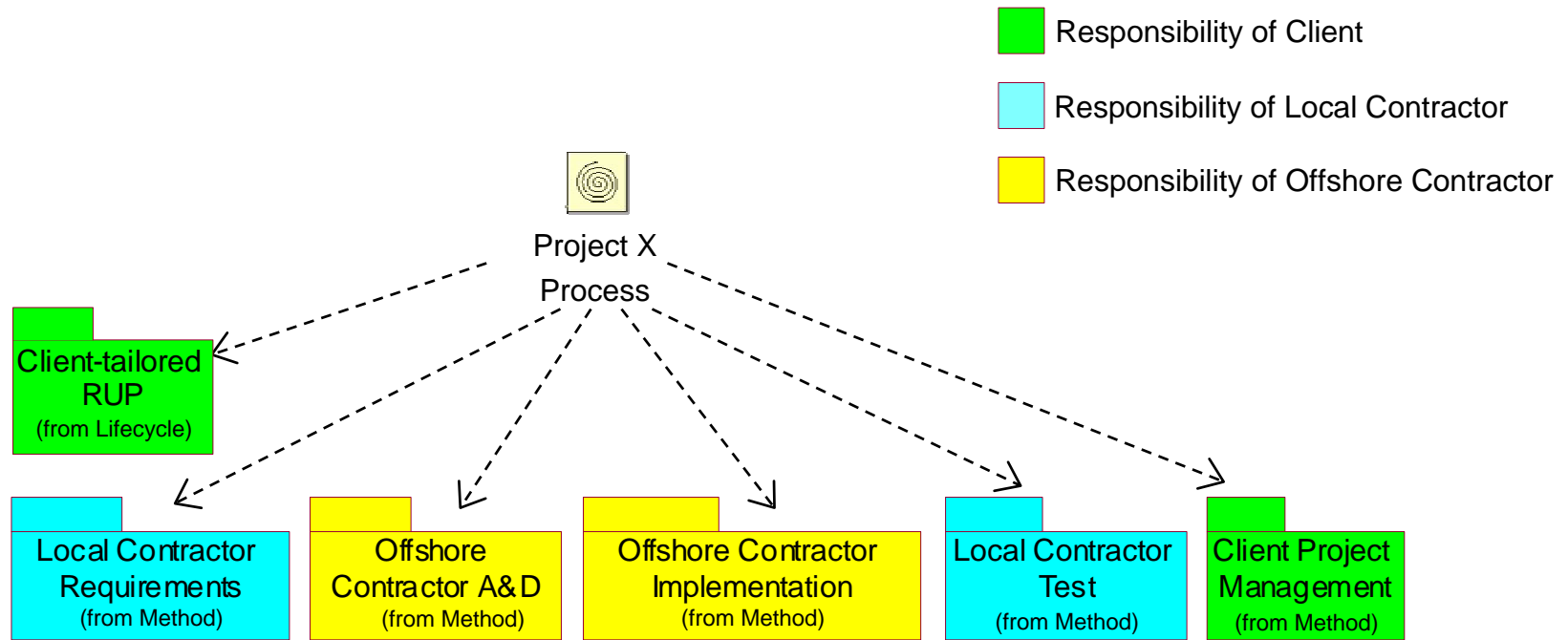
- MIL-STD-2167A Lifecycle with RUP Requirements**



Subcontracting Considerations



- **Client can define Lifecycle**
- **Methods can be dispersed amongst participating organizations**



- **Osellus IRIS**
 - **SPEM-compliant enterprise process automation application suite**
 - **Includes authoring, process execution, monitoring & measurement**
 - **Can create packages with PPA components**
 - **Can be used to assemble and publish process per PPA guidance**
 - **Does NOT require UML knowledge**

- **IBM Rational Process Workbench**
 - **SPEM-compliant process customization tool**
 - **Part of the RUP Platform**
 - **Can create UML packages with PPA components**
 - **Can be used to assemble and publish process per PPA guidance**
 - **Requires UML knowledge**

- **Reusable elements are too broad; we need additional constraints**
 - e.g. use-case design requires a scenario-based SRS
- **Need another class of “mixing” components**
 - e.g. Security (that involves requirements and test)
- **Strict adherence to the architecture impedes relationships across discipline methods**
 - Definition of impact relations amongst method-specific work products
 - Definition of precedence beyond what is implied by intermediary reusable work products

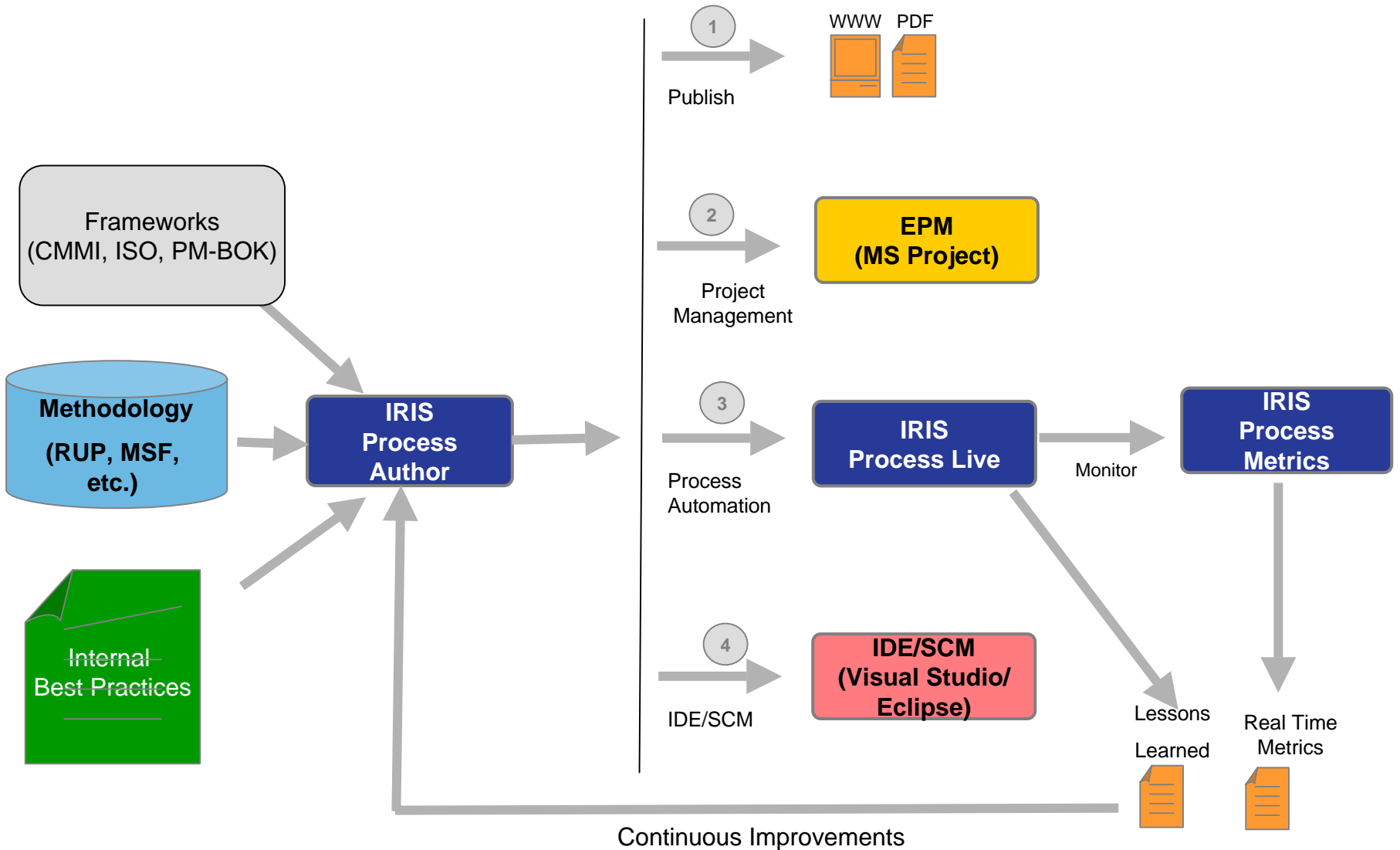
- **After all the technical talk, keep in mind the overarching idea:**
 - **An organization can define a process architecture with various lifecycles and methods**
 - **A concrete process can be defined by assembling these components based on a set of rules**
 - **That process will be complete**
 - **That process will be consistent**

Enabling PPA

Using IRIS Process Author

Sammy Wahab, Osellus Inc.

Process Automation with IRIS



End of Presentation

A Pluggable Process Architecture

For additional information please visit
<http://www.osellus.com/resources.html>

Or contact info@osellus.com

Instructions for Q&A

- **Press *1 to place yourself in the queue**
- **Ask the question when prompted**
- **If your question has already been asked or you want to remove yourself from the queue press #**