

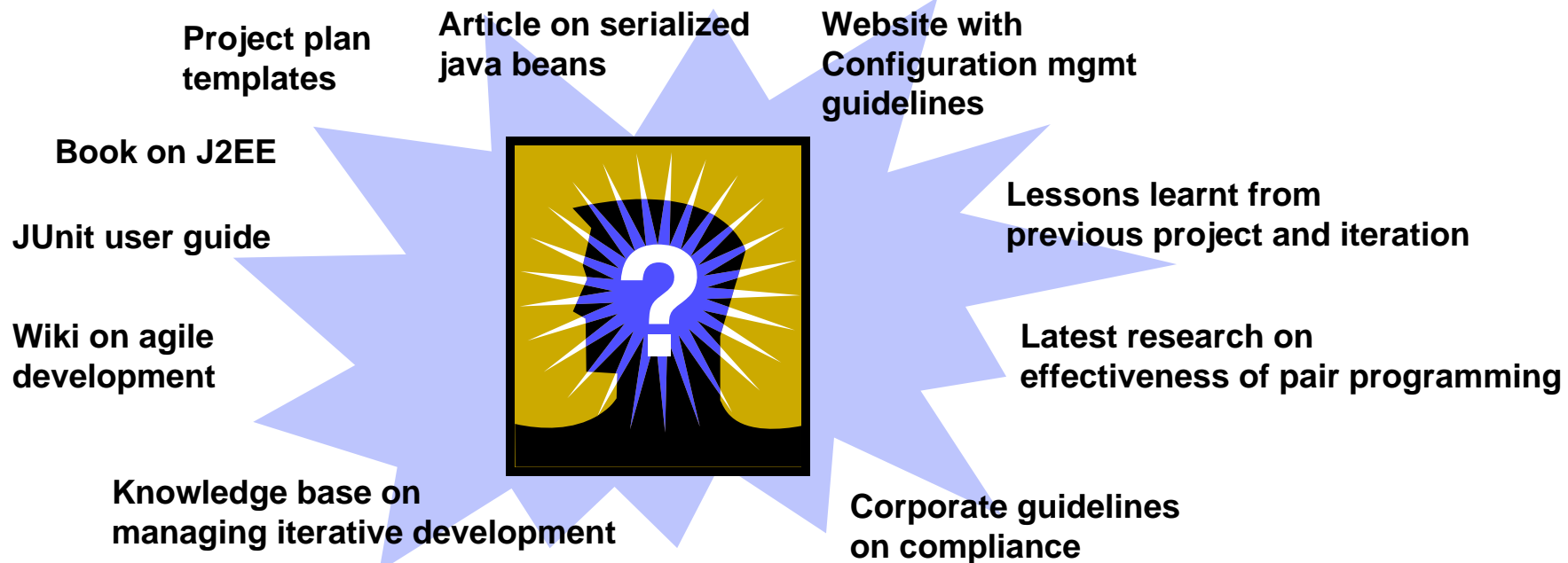
Introducing the Eclipse Process Framework

www.eclipse.org/epf

Peter Haumer
phaumer@us.ibm.com

EPF Project Committer

What Development Teams Are Facing Today



- No **common language** or terminology between processes - redundancy and inconsistencies
- Knowledge cannot easily be **customized** for different projects or new best practices
- No **central community** or **communication framework** to facilitate convergence of best practices across domains

A better approach

Standardize representation and manage libraries of reusable method content

Content on agile development

Content on managing iterative development

Guidance on serialized java beans



JUnit user guidance

Content on J2EE

Configuration mgmt guidelines



Develop and manage processes for performing projects

Lessons learnt from previous project and iteration

Corporate guidelines on compliance



Process assets patterns

Standard or reference processes

Project plan templates



Cohesive Web site customized for my project needs



Project plan templates and optional process instrumentation, relevant for the context of my project

Eclipse Process Framework Project Goals

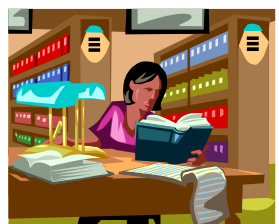
- Provide an **extensible framework** and **exemplary tools and content** for software process engineering
 - Extensible Framework
 - Meta-model based on OMG SPEM
 - Core extensible process tooling framework
 - Exemplary and extensible tools
 - Method and Process authoring
 - Library management and content extensibility
 - Configuring and publishing
 - Exemplary and extensible process content
 - Range of software development and management processes supporting
 - iterative, agile, and incremental development
 - applicable to a broad set of development platforms and applications

Value to the industry

- Converge industry process investments through a common process framework
- Broadly endorsable and will bring diverse process champions together
- A foundation for the industry to engineer, communicate and reuse process assets
- Tools, metamodel and core content for seeding an ecosystem
 - More reuse, more industry reach, more process/tool integration

Manage textual IC and reusable method content.

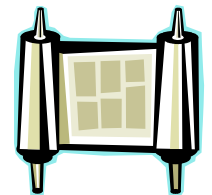
EPF Composer



IC / Best Practices



Books / Publications



Standards / Regulations



Homegrown Methods

Eclipse Process Framework Composer - C:\Home\Rational RUP team\TNG\Infrastructure\Samples\rup_20050708-playing

File Edit Search Configuration Window Help

config_for_PH_basic

Configuration X Content

config_for_PH_basic

- Requirements
 - Capture a Common Vocabulary
 - Detail a Use Case**
 - Detail the Software Requirement
 - Develop Requirements Management
 - Develop Vision
 - Elicit Stakeholder Requests
 - Find Actors and Use Cases
 - Manage Dependencies
 - Prioritize Use Cases
 - Review Requirements
 - Structure the Use-Case Model
- Test
- Uncategorized Tasks
- Domains
- Work Product Kinds
 - Code
 - Deliverables
 - Document
 - Environment Infrastructure
 - Management Data
 - Model
 - Analysis Model
 - Architectural Proof-of-Concept
 - Business Analysis Model
 - Business Use Case Model
 - Data Model
 - Deployment Model
 - Design Model
 - Implementation Model
 - Navigation Map
 - Software Requirement
 - Storyboard
 - Use-Case Model
 - Workload Analysis Model
 - Model Element
 - Plan
 - Record
 - Uncategorized
 - Role Sets
 - Roles
 - Managers
 - Analysts
 - Business Designer
 - Business-Process Analyst
 - Requirements Specifier
 - Stakeholder
 - System Analyst
 - Developers
 - General Roles
 - Production & Support
 - Testers

Task: Detail a Use Case

Discipline: Requirements

Expand All Sections Collapse All Sections

Purpose

- To describe one or more of the use case's flow of events in sufficient detail to enable software development to begin on it.
- To describe the use case specification to the understanding and satisfaction of the actor representative or customer.

Back to top

Relationships

Roles	Primary Performer: • Requirements Specifier	Additional Performers:
Inputs	Mandatory: • Use Case • Iteration Plan	Optional: • Glossary • Stakeholder Requests • Use-Case Model • Supplementary Specifications • Requirements Management Plan • Vision • Storyboard
Outputs	• Use Case • Supplementary Specifications	

Back to top

Steps

Expand All Steps Collapse All Steps

Review and Refine the Scenarios

Start by reviewing and refining the scenarios that you will be dealing with in the development cycle. These may have already been initially identified in the Task: Find Actors and Use Cases. Use these enumerated scenarios as a starting point in determining the scope of what flows will need to be described.

Storyboards will help you in understanding and detailing the use case flows. Another input to consider is the User-Interface Prototype, if one has already been developed.

- Detail the Flow of Events
- Structure the Flow of Events
- Illustrate Relationships with Actors and Other Use Cases
- Describe any Special Requirements

Easy to use, form-based rich-text editing capabilities.

Eclipse Process Framework Composer - C:\Home\Rational RUP team\TNG\Infrastructure\Samples\Beacon

File Edit Search Internal Configuration Window Help

Classic RUP (for large projects)

Library

- rup
 - Method Content
 - Content Packages
 - Architecture
 - Assessment
 - Design
 - Database Design
 - Design with Use-Case
 - GUI Design
 - Operation Design
 - Real-Time Design
 - Roles
 - Tasks
 - Work Products
 - rup_analysis_mod
 - rup_analysis_
 - rup_design_mode
 - Guidance

Configuration

Classic RUP (for large projects)

- Disciplines
- Domains
 - RUP Domains
 - Analysis and Design
 - Data Migration Specific
 - Analysis Model
 - Analysis Class
 - Architectural Proof-of-C
 - Data Model
 - Deployment Model
 - Design Model
 - Navigation Map
 - Reference Architecture
 - Software Architecture C
 - User-Interface Prototy
 - Service Model
 - Business Modeling
 - Configuration & Change Ma

rup_analysis_class

Work Product (Artifact): rup_analysis_class

General Information
Provide general information about this artifact.

Name: rup_analysis_class
Presentation name: Analysis Class
Unique ID:
Brief description: This work product specifies elements of an early conceptual model for 'things in the system which have responsibilities and behavior'.

Detail Information
Provide detailed information about this artifact.

Purpose: Analysis classes are used to capture the major system.

Main description: Analysis Classes specify elements of an early c the system which have responsibilities and beh prototypical classes of the system, and are a 'fir abstractions that the system must handle. Anal maintained in their own right, if a "high-level", co is desired. Analysis classes also give rise to the

Key considerations:

Notation
Provide notation information about this artifact.

Brief outline:

Representation options:

Tailoring
Provide tailoring information about this artifact.

Impact of not having:

Work Product (Artifact): rup_analysis_class

Representation options:

CRC Card technique - see [WIR90] for details of this technique). On the front side of the card, capture the name and description of the class. An example for a Course in a course registration system is listed below:

Class Name	Course									
Description	The Course is responsible for maintaining information about a set of course sections having a common subject, requirements and syllabus.									
Responsibilities	To maintain information about the course.									
Attributes	<table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Type</th> </tr> </thead> <tbody> <tr> <td>Course Title</td> <td>The name of the course</td> <td>string</td> </tr> <tr> <td>Description</td> <td>A short description of the course</td> <td>string</td> </tr> </tbody> </table>	Name	Description	Type	Course Title	The name of the course	string	Description	A short description of the course	string
	Name	Description	Type							
Course Title	The name of the course	string								
Description	A short description of the course	string								

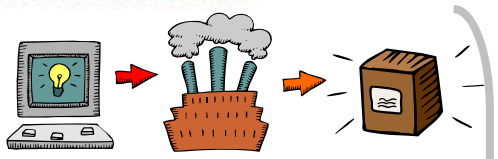
On the back of the card, draw a diagram of the class:

```

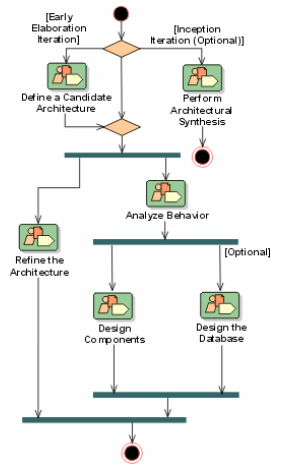
classDiagram
    class Course {
        name
        description
    }
    class Section
    class Professor
    class Student
    class Textbook
    class Room

    Course "0..*" -- "0..*" Section
    Section "1..*" -- "1..*" Professor
    Section "1..*" -- "1..*" Student
    Section "1..*" -- "1..*" Textbook
    Section "1..*" -- "1..*" Room
    
```

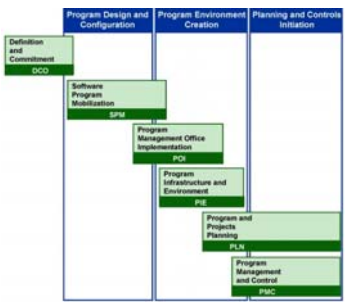
Enable processes by with reusable method content and process patterns. Consistent management of WBS and Activity diagrams.



Workflow



Workflow



Breakdown / Iterations / Increments / Sprints / Waterfall

Eclipse Process Framework Composer - C:\Home\Rational RUP team\TNG\Infrastru...

File Edit Search Configuration Window Help

config_for_PH_basic

Library

- basic_unified_process
 - Method Content
 - Processes
 - PH_basic_business_integration
 - RUP extension

Configuration

config_for_PH_basic

- Capture a Common Vocabulary
- Detail a Use Case
- Detail the Software Requirements
- Develop Requirements Management
- Develop Vision
- Elicit Stakeholder Requirements
- Find Actors and Use Cases
- Manage Dependencies
- Prioritize Use Cases
- Review Requirements
- Structure the Use-Case I
- Test
- Uncategorized Tasks
- Domains
 - Analysis and Design
 - Business Modeling
 - Configuration & Change Management
 - Deployment
 - Environment
 - Implementation
 - Project Management
 - Requirements
 - Glossary
 - Requirements Attributes
 - Requirements Management
 - Software Requirement
 - Software Requirements
 - Stakeholder Requests
 - Storyboard
 - Supplementary Specifications
 - Use-Case Model
 - Vision
 - Test
 - Uncategorized
 - Work Product Kinds
 - Role Sets
 - Roles
 - Managers
 - Analysts
 - Business Designer

Small Use Case-based Process

Presentation Name	Index	Model Info	Type	Pred
Small Use Case-based Process	0		Capability P...	
Inception	1		Phase	
Understand Stakeholder Needs	2		Activity	
Find Actors and Use Cases	3		Task Descri...	
Define the System	4		Activity	2
Detail a Use Case	5		Task Descri...	
System Analyst		Primary Performer	Role Descrip...	
Requirements Specifier		Secondary Performer	Role Descrip...	
Use Case		Mandatory Input	Artifact Des...	
Glossary		Optional Input	Artifact Des...	
Use-Case Model		Optional Input	Artifact Des...	
Storyboard		Optional Input	Artifact Des...	
Stakeholder Requests		Optional Input	Artifact Des...	
Use Case		Output	Artifact Des...	
Prioritize Use Cases	6		Task Descri...	
Other Work...	7		Activity	4
Elaboration	8		Phase	
Refine the System Definition	9		Activity	
Detail a Use Case	10		Task Descri...	
Requirements Specifier		Primary Performer	Role Descrip...	
Use Case		Mandatory Input	Artifact Des...	
Iteration Plan		Mandatory Input	Artifact Des...	
Glossary		Optional Input	Artifact Des...	
Stakeholder Requests		Optional Input	Artifact Des...	
Use-Case Model		Optional Input	Artifact Des...	
Supplementary Specifications		Optional Input	Artifact Des...	
Requirements Management Plan		Optional Input	Artifact Des...	
Vision		Optional Input	Artifact Des...	
Storyboard		Optional Input	Artifact Des...	
Use Case		Output	Artifact Des...	
Supplementary Specifications		Output	Artifact Des...	
Analyze Behaviour	11		Activity	
Use-Case Analysis	12		Task Descri...	

Pha...

stand Stakeholder

Define the System

Other Work...

Description | Work Breakdown Structure | Team Allocation | Work Product Usage | Consolidated View

Properties

Task Descriptor : detail_a_use_case

General

Documentation

Roles

Work Products

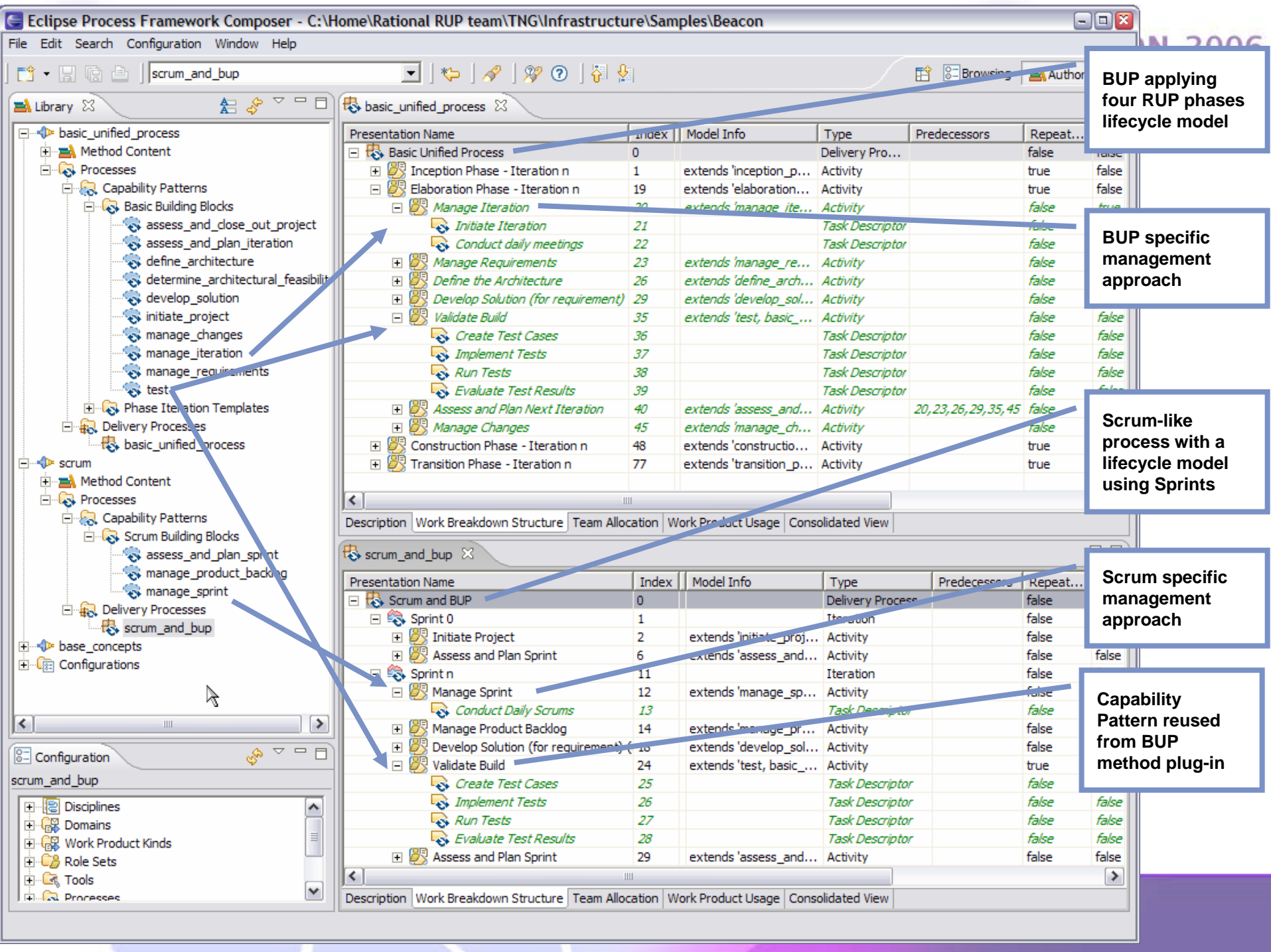
Steps

Specify the steps to perform this task descriptor.

Steps:

- Review and Refine the Scenarios
- Detail the Flow of Events
- Illustrate Relationships with Actors and Other Use Cases
- Evaluate Your Results

Add...



BUP applying four RUP phases lifecycle model

BUP specific management approach

Scrum-like process with a lifecycle model using Sprints

Scrum specific management approach

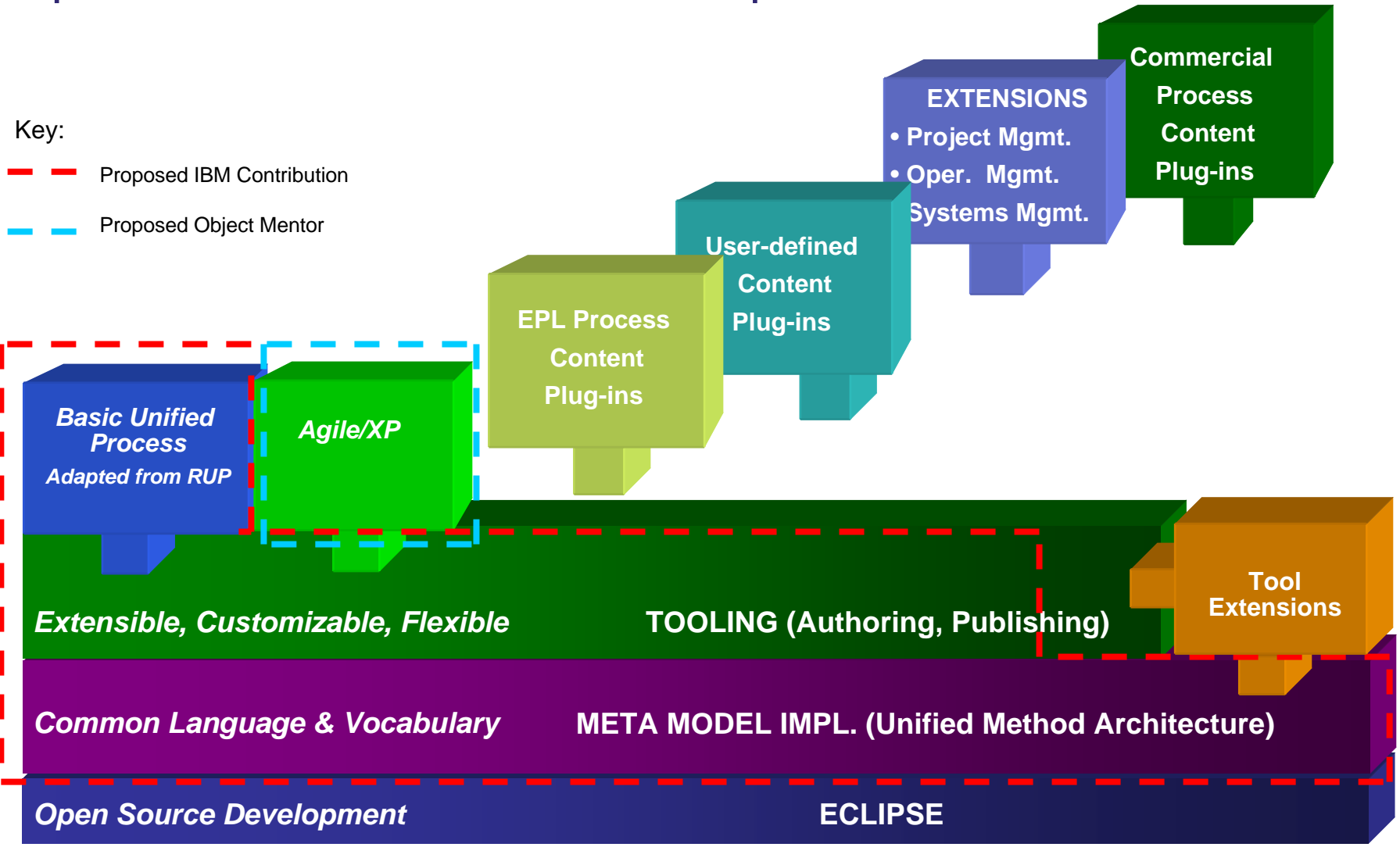
Capability Pattern reused from BUP method plug-in

- basic_unified_process
 - Method Content
 - Processes
 - Capability Patterns
 - Basic Building Blocks
 - assess_and_close_out_project
 - assess_and_plan_iteration
 - define_architecture
 - determine_architectural_feasibility
 - develop_solution
 - initiate_project
 - manage_changes
 - manage_iteration
 - manage_requirements
 - test
 - Phase Iteration Templates
 - Delivery Processes
 - basic_unified_process
- scrum
 - Method Content
 - Processes
 - Capability Patterns
 - Scrum Building Blocks
 - assess_and_plan_sprint
 - manage_product_backlog
 - manage_sprint
 - Delivery Processes
 - scrum_and_bup
- base_concepts
- Configurations

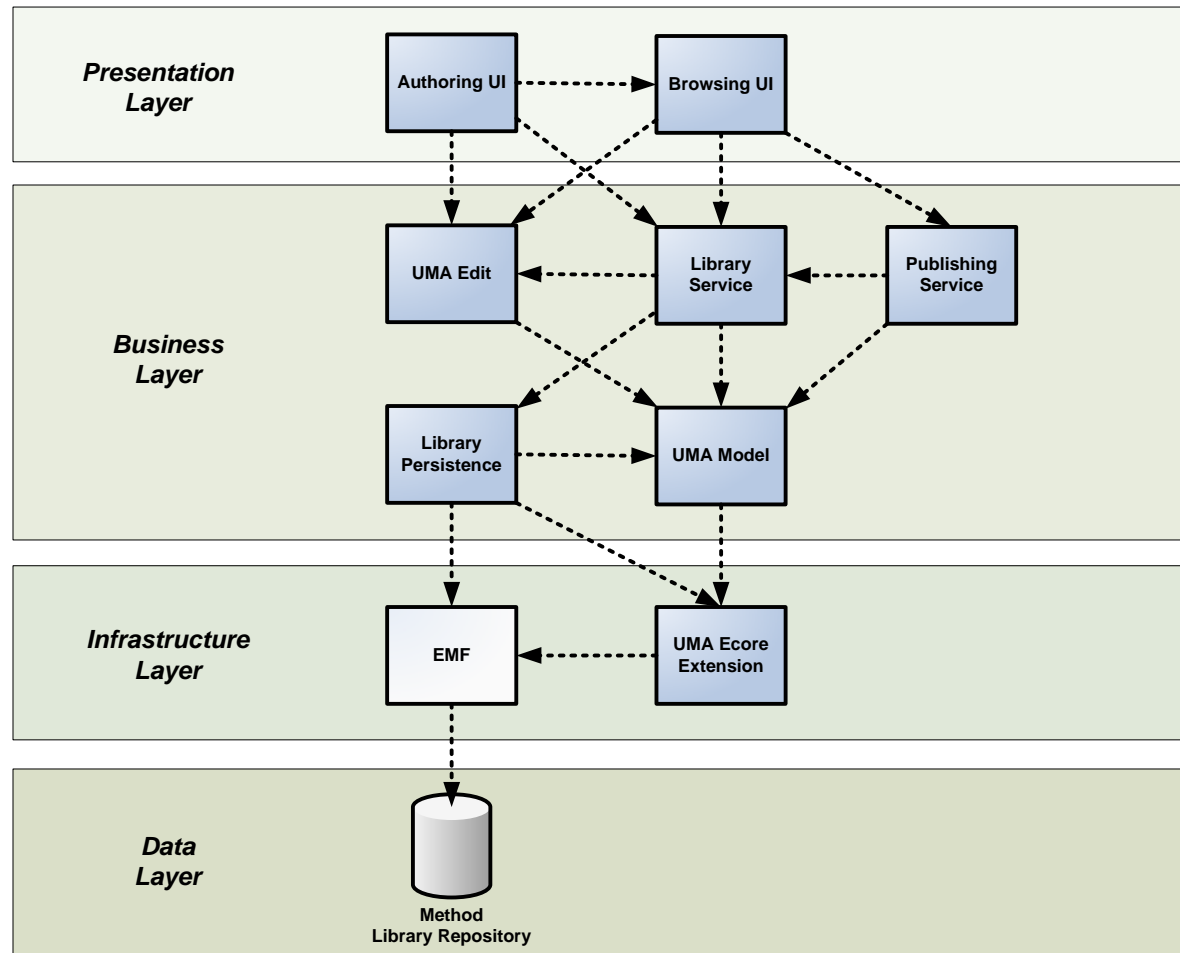
Presentation Name	Index	Model Info	Type	Predecessors	Repeat...
Basic Unified Process	0		Delivery Pro...		false
Inception Phase - Iteration n	1	extends 'inception_p...	Activity		true
Elaboration Phase - Iteration n	19	extends 'elaboration...	Activity		true
Manage Iteration	20	extends 'manage_ite...	Activity		false
Initiate Iteration	21		Task Descriptor		false
Conduct daily meetings	22		Task Descriptor		false
Manage Requirements	23	extends 'manage_re...	Activity		false
Define the Architecture	26	extends 'define_arch...	Activity		false
Develop Solution (for requirement)	29	extends 'develop_sol...	Activity		false
Validate Build	35	extends 'test, basic_...	Activity		false
Create Test Cases	36		Task Descriptor		false
Implement Tests	37		Task Descriptor		false
Run Tests	38		Task Descriptor		false
Evaluate Test Results	39		Task Descriptor		false
Assess and Plan Next Iteration	40	extends 'assess_and...	Activity	20,23,26,29,35,45	false
Manage Changes	45	extends 'manage_ch...	Activity		false
Construction Phase - Iteration n	48	extends 'constructio...	Activity		true
Transition Phase - Iteration n	77	extends 'transition_p...	Activity		true

Presentation Name	Index	Model Info	Type	Predecessors	Repeat...
Scrum and BUP	0		Delivery Process		false
Sprint 0	1		Iteration		false
Initiate Project	2	extends 'initiate_proj...	Activity		false
Assess and Plan Sprint	6	extends 'assess_and...	Activity		false
Sprint n	11		Iteration		false
Manage Sprint	12	extends 'manage_sp...	Activity		false
Conduct Daily Scrums	13		Task Descriptor		false
Manage Product Backlog	14	extends 'manage_pr...	Activity		false
Develop Solution (for requirement)	18	extends 'develop_sol...	Activity		false
Validate Build	24	extends 'test, basic_...	Activity		true
Create Test Cases	25		Task Descriptor		false
Implement Tests	26		Task Descriptor		false
Run Tests	27		Task Descriptor		false
Evaluate Test Results	28		Task Descriptor		false
Assess and Plan Sprint	29	extends 'assess_and...	Activity		false

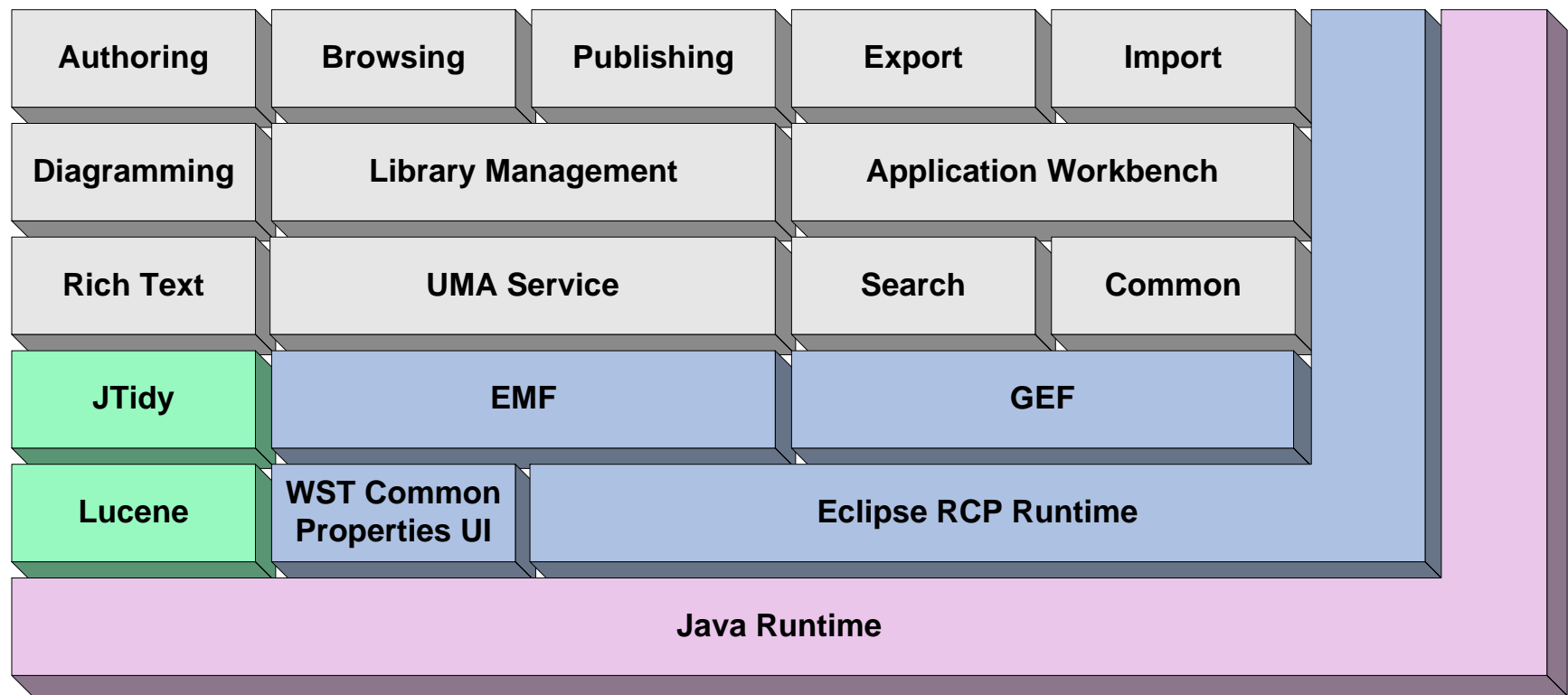
Eclipse Process Framework Conceptual Architecture



Beacon/RMC Runtime Architecture



High-Level Architecture



- EPF tool components
- Eclipse components
- Other open source components
- Sun/IBM JRE

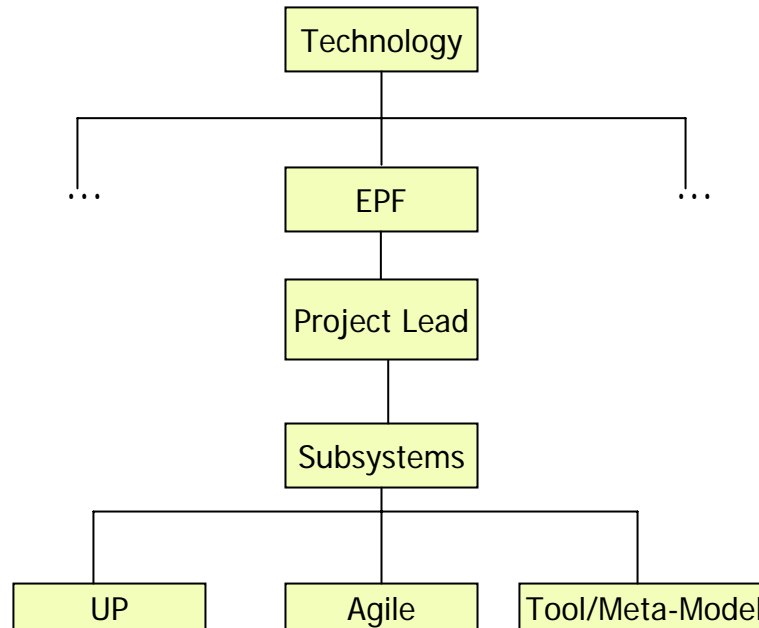
Technology Stack

- Eclipse Rich Client Platform (RCP) 3.1
- Eclipse Modeling Framework (EMF) 2.1
- XML Meta Interchange (XMI) 2.0
- Extensible Stylesheet Language Transformation (XSLT) 1.0
- XML Localization Interchange File Format (XLIFF) 1.0
- UML 2.0 Diagram Interchange Specification
- Graphical Editing Framework (GEF) 3.1
- Java2D
- Eclipse Forms
- DHTML (HTML, JavaScript, DOM & CSS)
- W3C Jtidy
- Apache Lucene
- Java Applet

Industry Support

- IBM
- 2-Pro Mentor
- Adaptive
- Ambysoft
- Armstrong Process Group
- BearingPoint
- Bedarra Research Labs
- Capgemini
- Catalysts
- Covansys
- Ivar Jacobson International
- Number Six Software
- Object Mentor
- SOFTEAM
- University of British Columbia
- aubryConseil
- European Software Institute
- Jaczone
- NTT Comware
- Object Management Group (OMG)
- Osellus
- Sogeti
- Telelogic
- Unisys
- Wind River

EPF Project Structure



EPF Project Timeline

