

# Agenda

## Context

- Capability Engineering Process (CEP)
- Patterns and Agility for Capability Engineering Methodologies(PACEM)

## Process Patterns

- Agility & Process Patterns

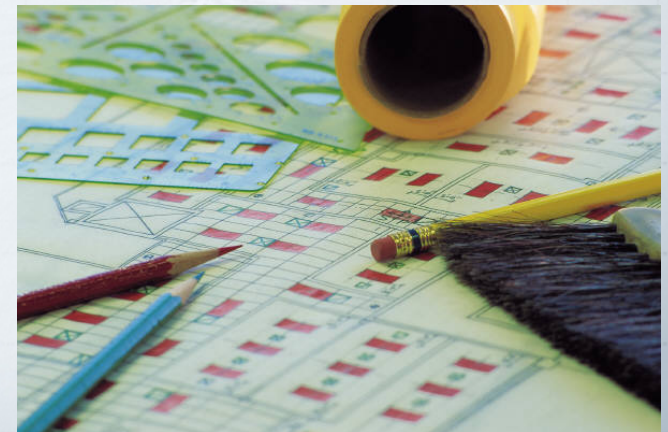
## Experimentation

- Incorporating agility within capability engineering through process patterns

## Way ahead

- Assessment through simulation or experimentation

# CONTEXT





# Defence R&D Canada

**Mission :** *To ensure the Canadian Forces are technologically prepared and operationally relevant.*

- Advise on Science & Technology
- Conduct defence research, development and analysis
- Assess technology trends, threats, and opportunities
- Engage industrial, academic and international partners in the transition of technology
- Conduct S&T projects for non-DND clients





# CapDEM project

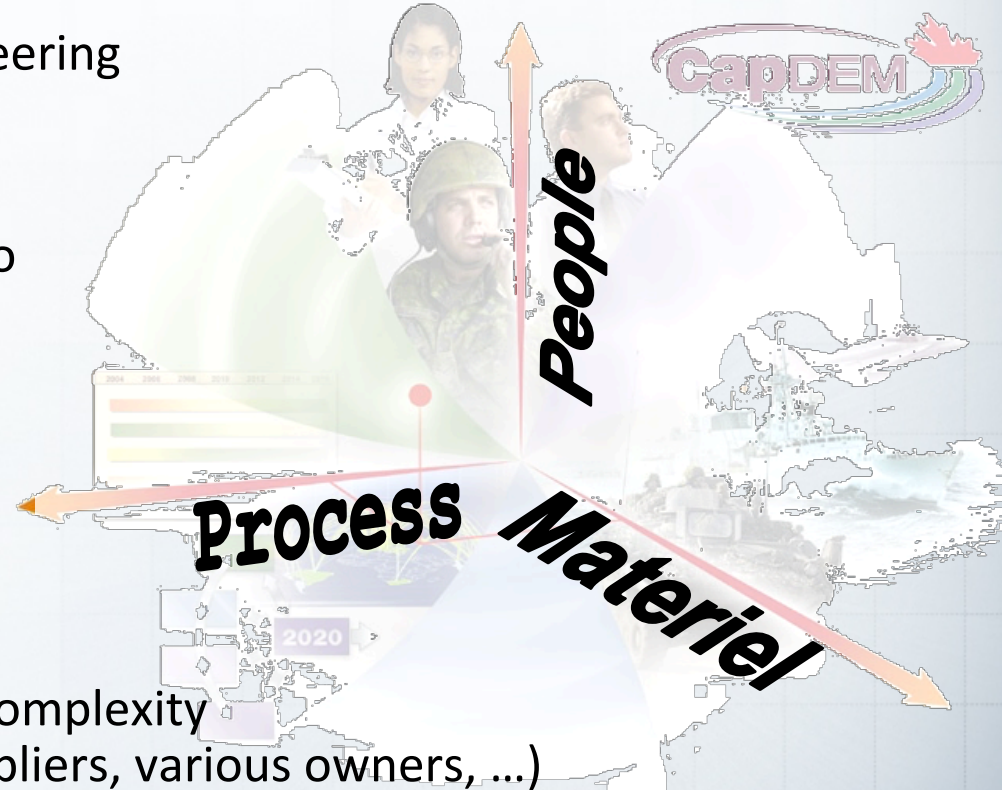
## Collaborative cAPability Definition, Engineering and Management

- Major R&D effort 2004-2009
- To transpose rigour of system engineering to capability/system-of-systems engineering

Delivering strategic options to executive decision makers

Tested with use cases  
(Alpha, Beta, Gamma)

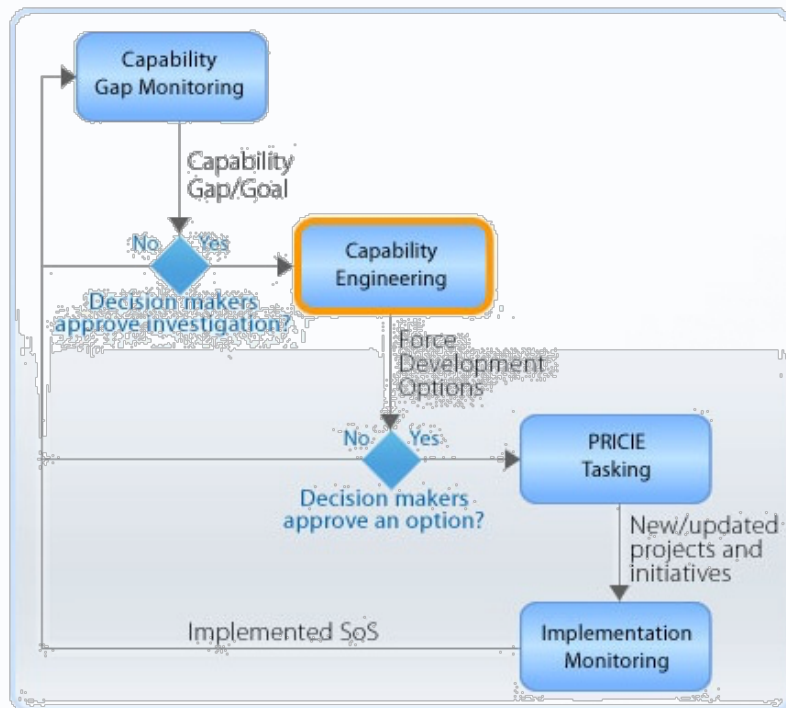
Dealing with complexity  
(many systems, different suppliers, various owners,...)



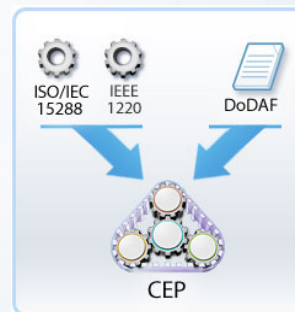




# Goal & Context

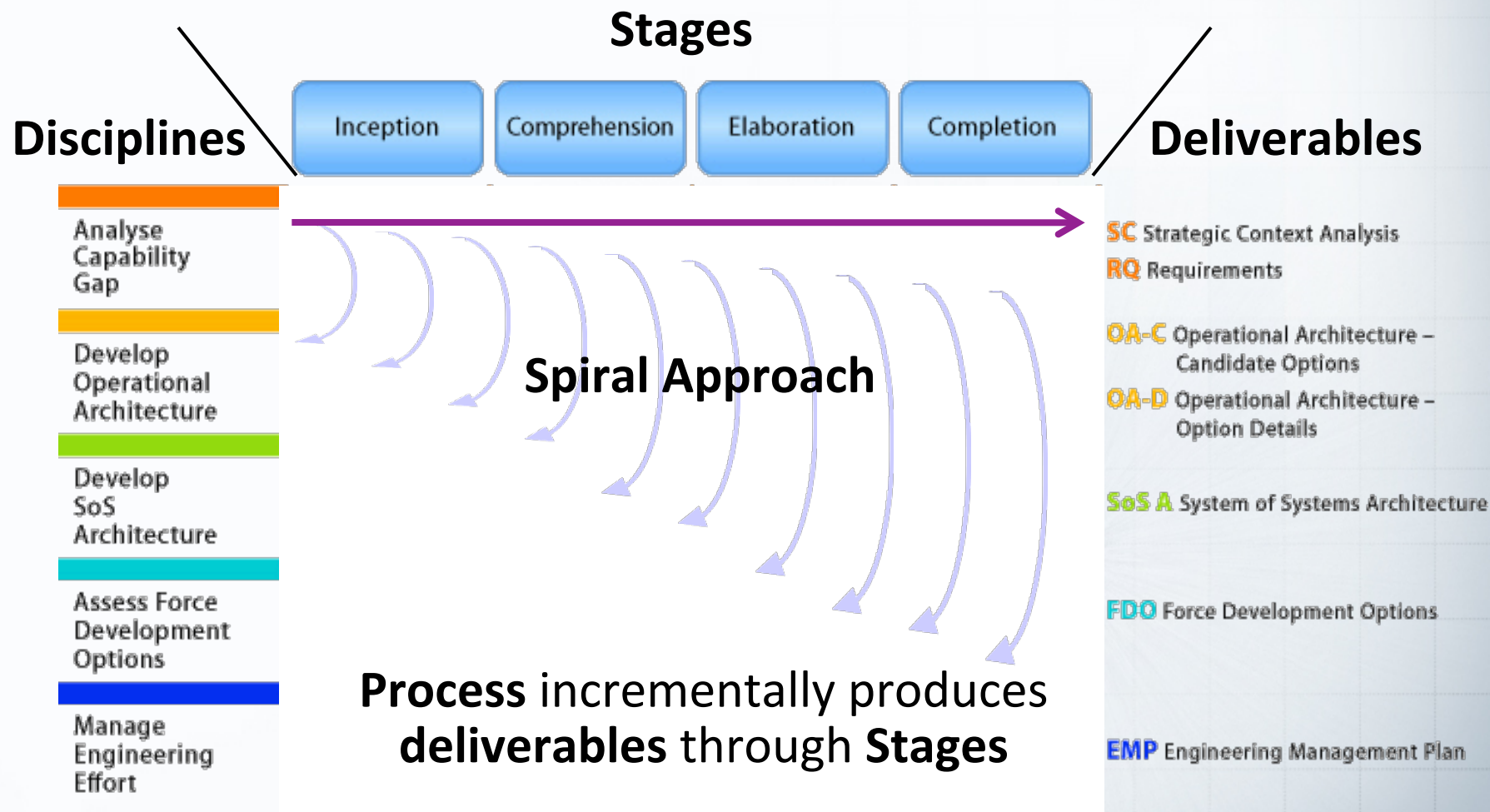


- From a capability shortcoming statement
- Explore potential options
- Issue a « Capability » recommendation
  - Justified in term of performance, cost, schedule, risk
  - With the option documentation
    - Architecture (operational and System)
    - Requirements
  - Articulated around the PRICIE elements



P	• Personnel
R	• R&D
I	• Infrastructure
C	• Concept & Doctrine
I	• Information management
E	• Equipment & Supplies

# Capability Engineering Process (CEP)





## CEP 2011

## CEP 2011

- CEP 2011
  - CEP Overview
    - Background
    - Foundations
  - Perspectives
    - Global Perspective
    - Perspective on Timeline
    - Perspective on Disciplines
    - Perspective on Deliverables
  - Workflow
    - Overview
    - Inception
      - Define Decision Criteria
      - Describe Operational Concept
      - Describe Relevant Systems
      - Describe Strategic Facets
      - Evaluate Opportunity
      - Manage Iteration
      - Organize Next Iteration
      - Setup Initiative
      - Support Engineering Effort
      - Verify CEDF Inputs
    - Comprehension
    - Elaboration
    - Completion
  - Capability Engineering Team
  - CEP Deliverables
    - Engineering Management Plan
    - Force Development Options
    - Operational Architecture - Candidate Option
    - Operational Architecture - Option Details
    - Progress Report
    - Requirements
    - SoS Architecture
    - Strategic Context Analysis
  - CEP Artefacts
  - Example
  - Glossary/Acronyms
  - Bibliography & References
  - About

## CEP 2011

## CEP 2011



This presents the CEP 2011 in a glance.

[Expand All Sections](#) [Collapse All Sections](#)

## Main Description

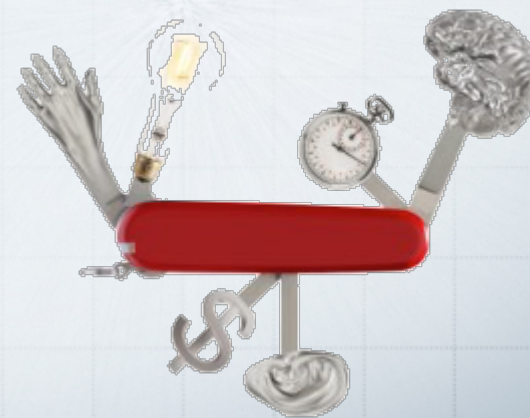

[Back to top](#)



# The PACEM project starting point

## Patterns & Agility for Capability Engineering Methodologies

- CEP = 1 “delivery process”
  - One workflow, one size fits all
- A versatile process
  - Various domains of application
  - Various usages
  - Various project environments
- We need **agile** processes & tools.



# Agility

Middle French, circa 1581.

from Latin agilis, agere : to drive, lead, act, do.

from Greek agein : to drive, lead.

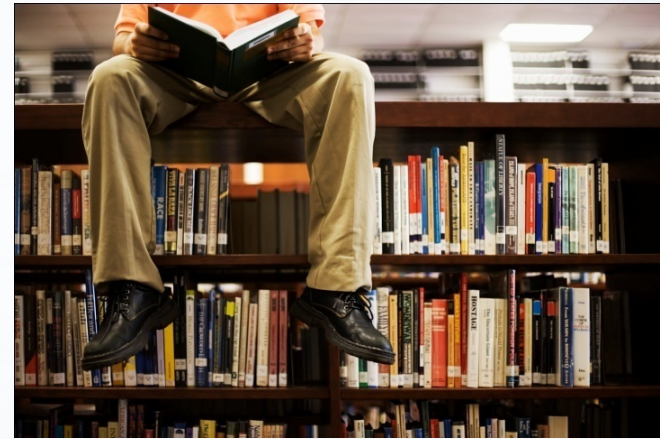
Other proto-Indo-European roots include :

ǵyṵ (agō) : to lead, (ancient Greek)

aka: to travel in a vehicle (Old Norse),

ag: Battle (Middle Irish)

अजति (ájati): to drive, propel, throw, cast. (Sanskrit)



**“Agility is the *ability* to *respond* to *changing circumstances* where:**

- **Ability** is characterized by readiness and speed of action
- **Response** is:
  - Making use of an existing configuration (by internal means)
  - By reconfiguration (facilitated by external means)
- **Changing circumstances** may be:
  - A change in objective
  - A change in environment
  - A change in condition
- **Agility** is measured by:
  - Speed of action
  - Cost in resource
  - Impact on effectiveness”.

Mackley, T., S. Barker, and P. John. 2008. Concepts of Agility in Network Enabled Capability. In RNEC Conference 2008. Leeds, UK

## Hint :

Patterns are successfully used in software engineering for increased agility.

Can we borrow this approach for our needs ?





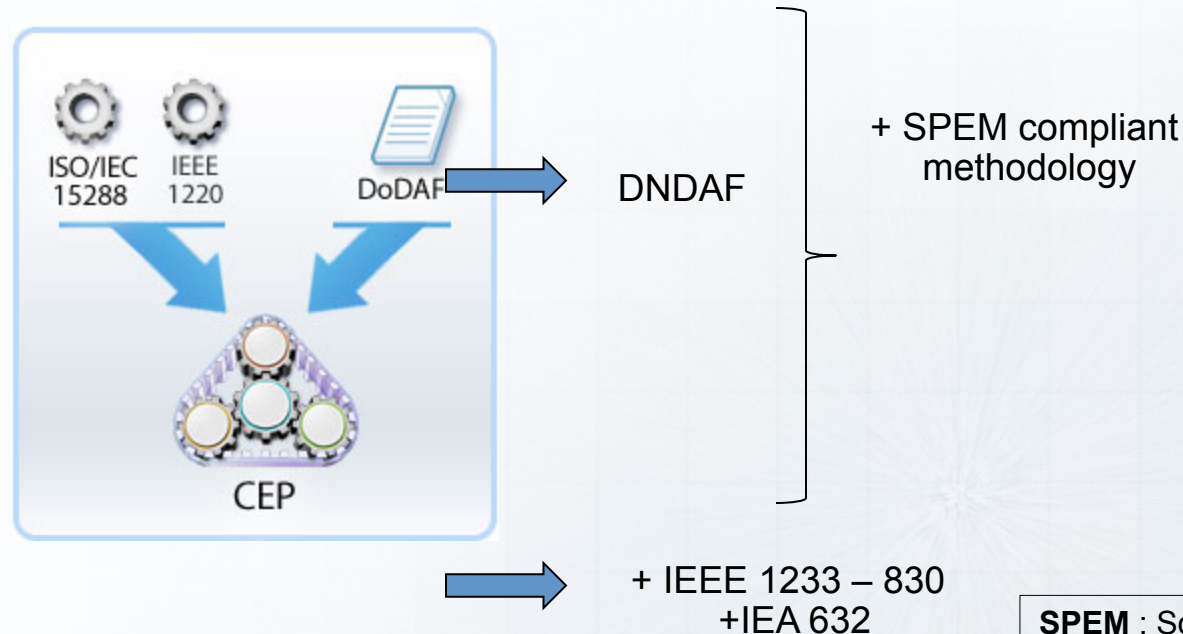
## PACEM Goals :

- Upgrade the actual capability engineering process
- Implement the concept of “Process patterns” within the CEP.
  - Identify & implement process patterns
  - Measure and assess process agility improvement resulting from patterns.



## From CEP to PACEM :

CEP : is an engineering process to recommend force development options with the support of architectures



PACEM : is an agile CEPs generator..

**SPEM** : Software  
Process Engineering  
Metamodel  
**DoDAF** : Department of  
Defense Architecture  
Framework  
**DNDAF** : Department of  
National Defence  
Architecture Framework

# Process Engineering, a Research Theme :

## Excerpts from the “Process Research Framework” :

- **How to define and build processes and understand their performance?**
- **E3 Providing process engineering infrastructure**
  - *Determine infrastructure needed to best support this process engineering environment*
- **E2 Organizing processes for reuse**
  - *Engineer, assemble, combine and reuse process components to meet performance targets*
- **E1 Specifying processes using evidence**
  - *Specify processes with adequate empirical evidence of their performance*

### “A Process Research Framework”

The International Process Research Consortium, Dec. 2006

[www.sei.cmu.edu/iprc](http://www.sei.cmu.edu/iprc)



From Patterns to Process Patterns

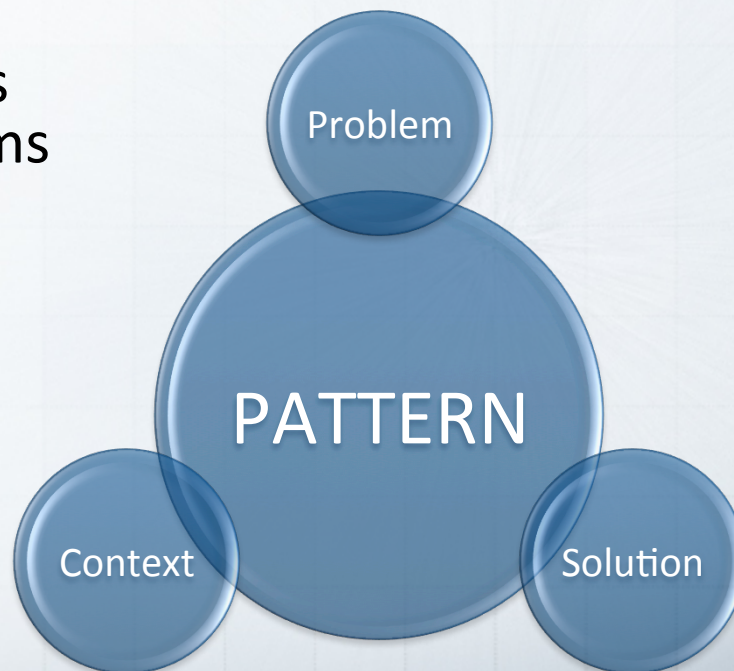
- Definition, examples and application.

# PROCESS PATTERNS



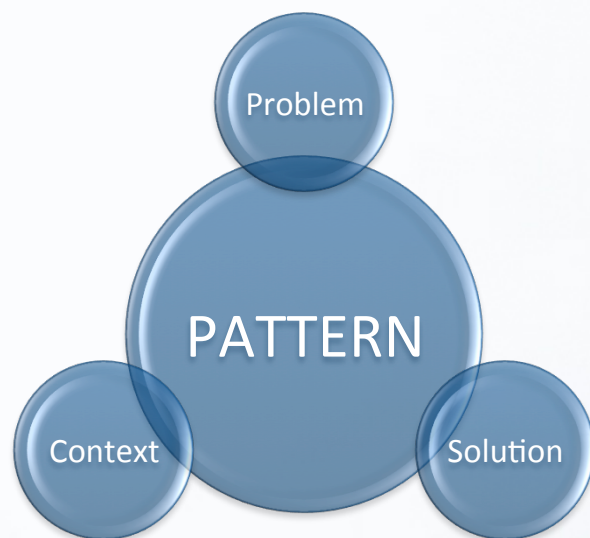
# Origins

- Christopher Alexander, the “father” of modern pattern concept :
  - “Each pattern is a three-part rule, which express a relation between a certain context, a problem and a solution”
  - Alexander C., 1979, “**The Timeless Way of Building**”, Oxford University Press, New York
- Patterns are generic solutions adaptable to specific problems
- Patterns are not invented but discovered



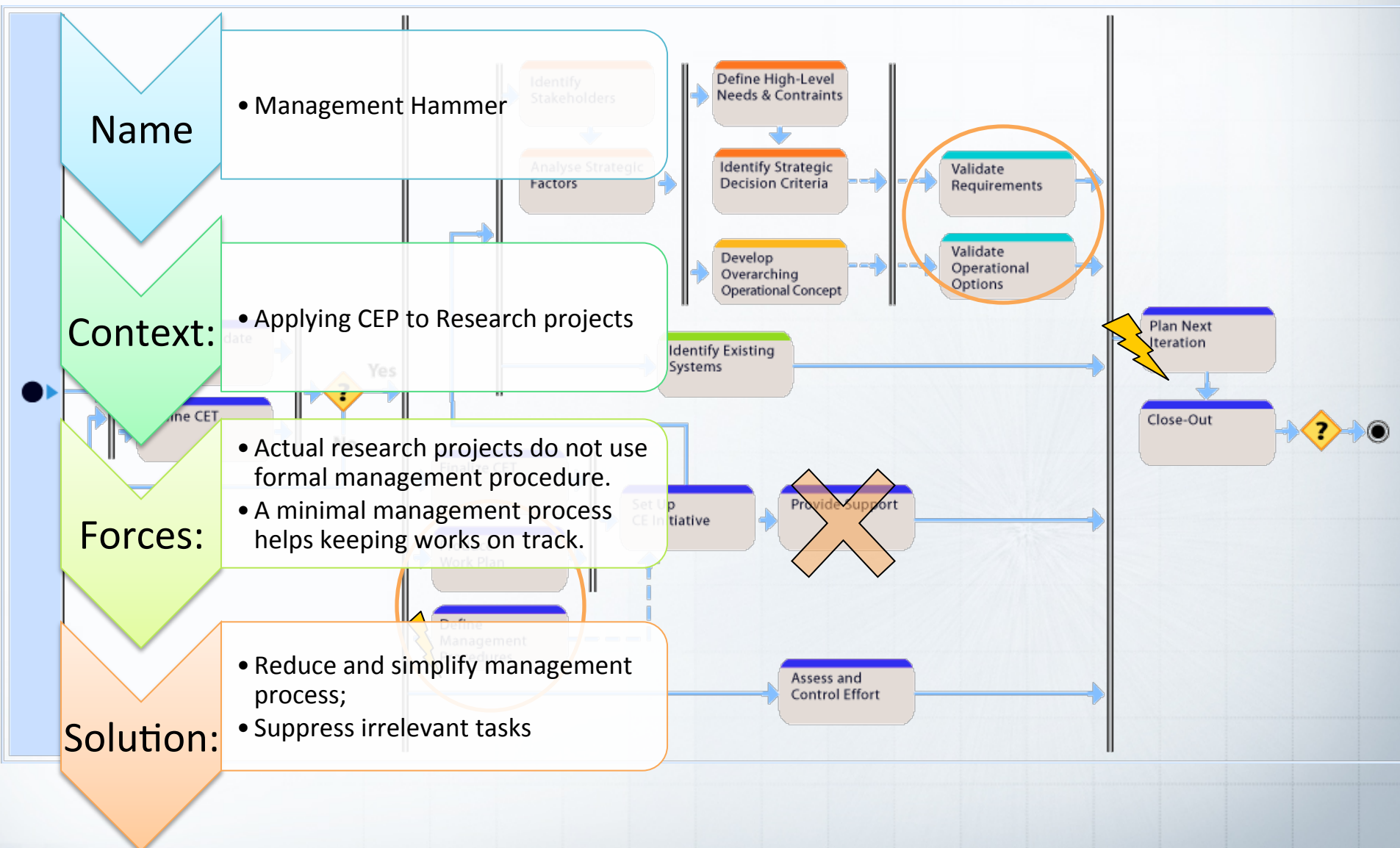
# Process patterns

- Process patterns are patterns which solution part applies to process.
- Process patterns describe techniques, tasks and activities used to design or adapt workflows, according to the project context and problems to be solved.



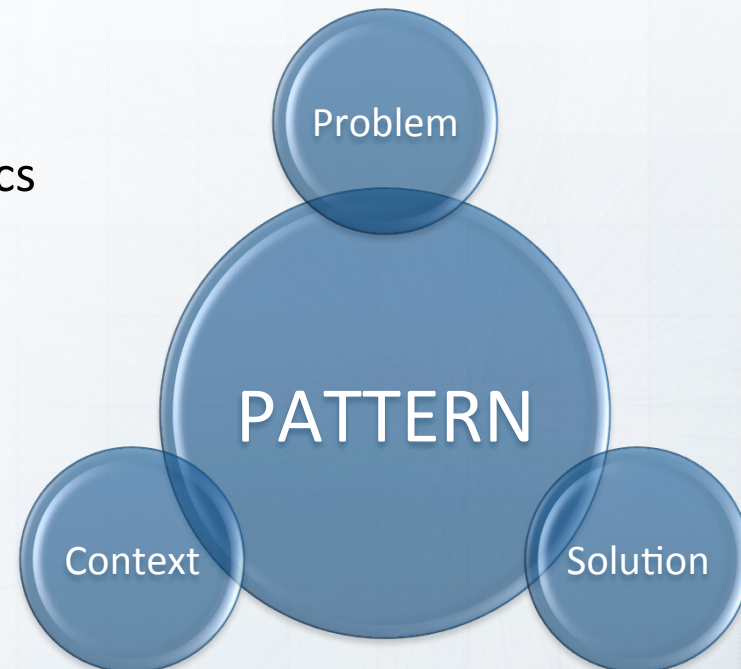


# Process Pattern example



# Process patterns documentation template

- General Info
  - Name, brief description, purpose, related patterns
- Context
  - Scope
  - Team characteristics
  - Project of concerns characteristics
- Problem
  - Description
  - Forces in presence
- Solution
  - Solution Description, Schema
  - Resulting context
  - Rationale
  - Known uses



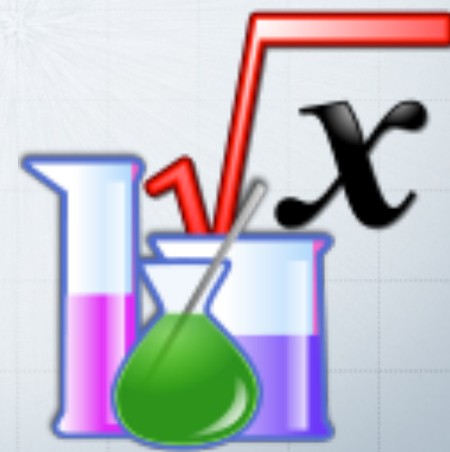
Incorporating agility within capability engineering through process patterns

Process engineering design environment

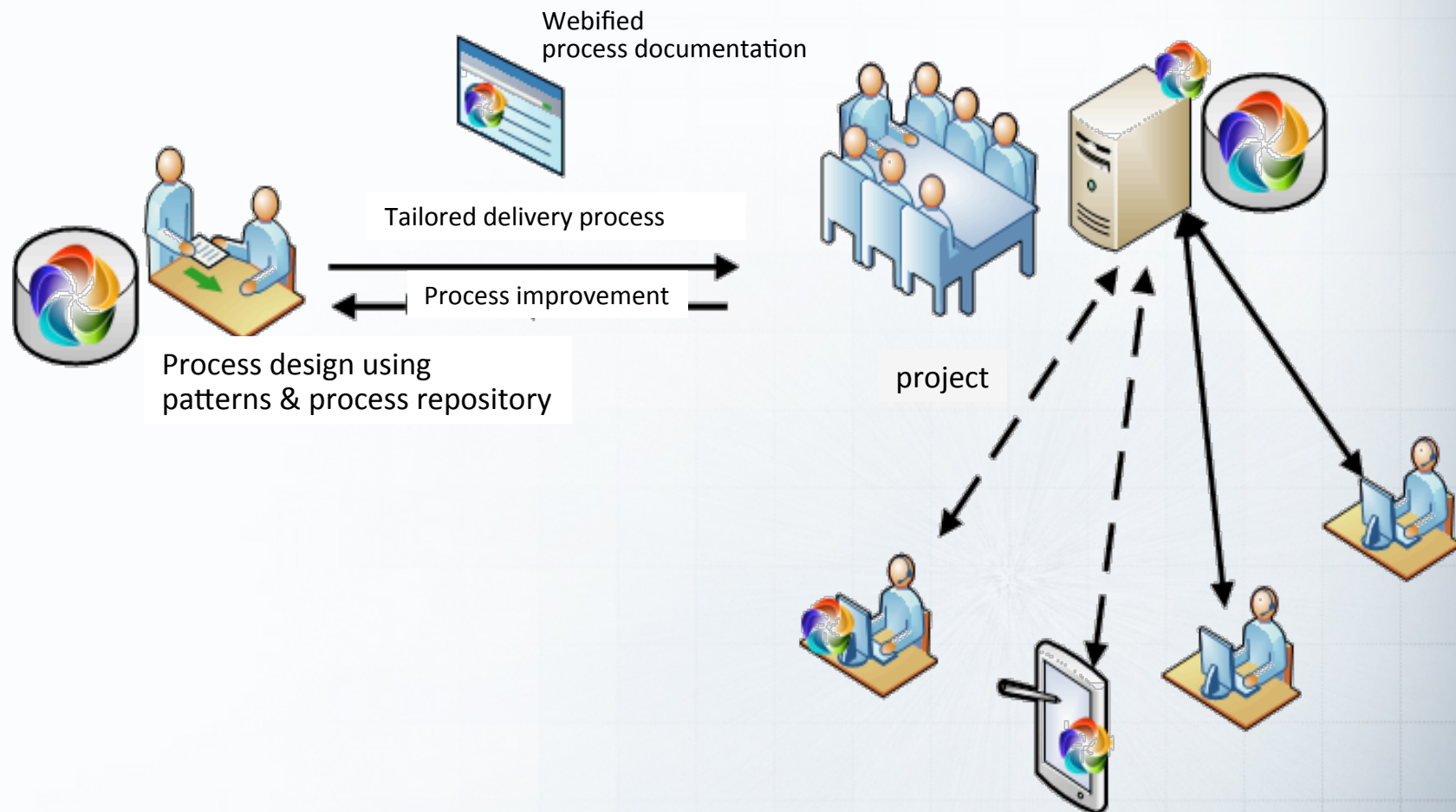
Process pattern identification

The High-Spin Pattern

# THE EXPERIMENTATION



# ALADIN : Integrated Capability Engineering Process Framework



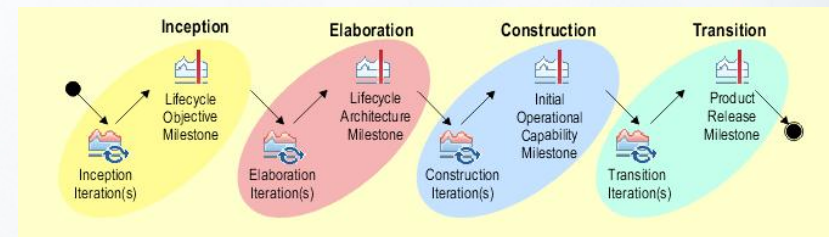
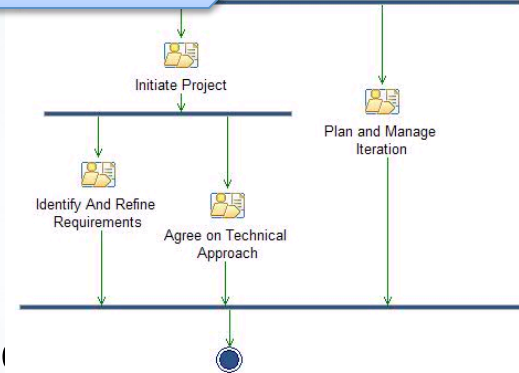


# Features and Tools

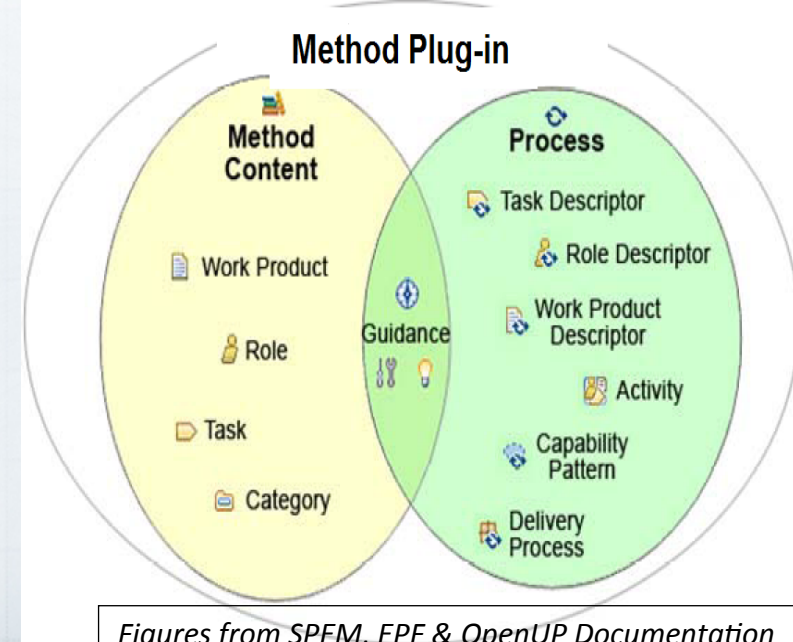
- Documentation
  - To constitute a repository of patterns and processes
    - Aladin, based on Eclipse Process Framework
- Simulation
  - To validate process agility
- Execution
  - To deploy and permit process execution
    - Using tools such as YAWL, Bonita, ...

# Related Technologies

- Process documentation standard
  - SPEM (Software Process Engineering Metamodel)



- OpenSource environment
  - Eclipse
    - EPF (Process Framework)
  - SVN (Deliverable repository)

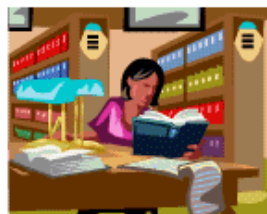


Figures from SPEM, EPF & OpenUP Documentation

# Eclipse Process Framework

- A design environment to
  - Document method content;
  - Design reusable process chunks;
  - Publish (web) customized processes.
- Eclipse based.
- Standard compliant :
  - UMA (Unified Method Architecture)
  - SPEM (Software Process Engineering Meta-model)

# Aladin : Method authoring



IC / Best Practices



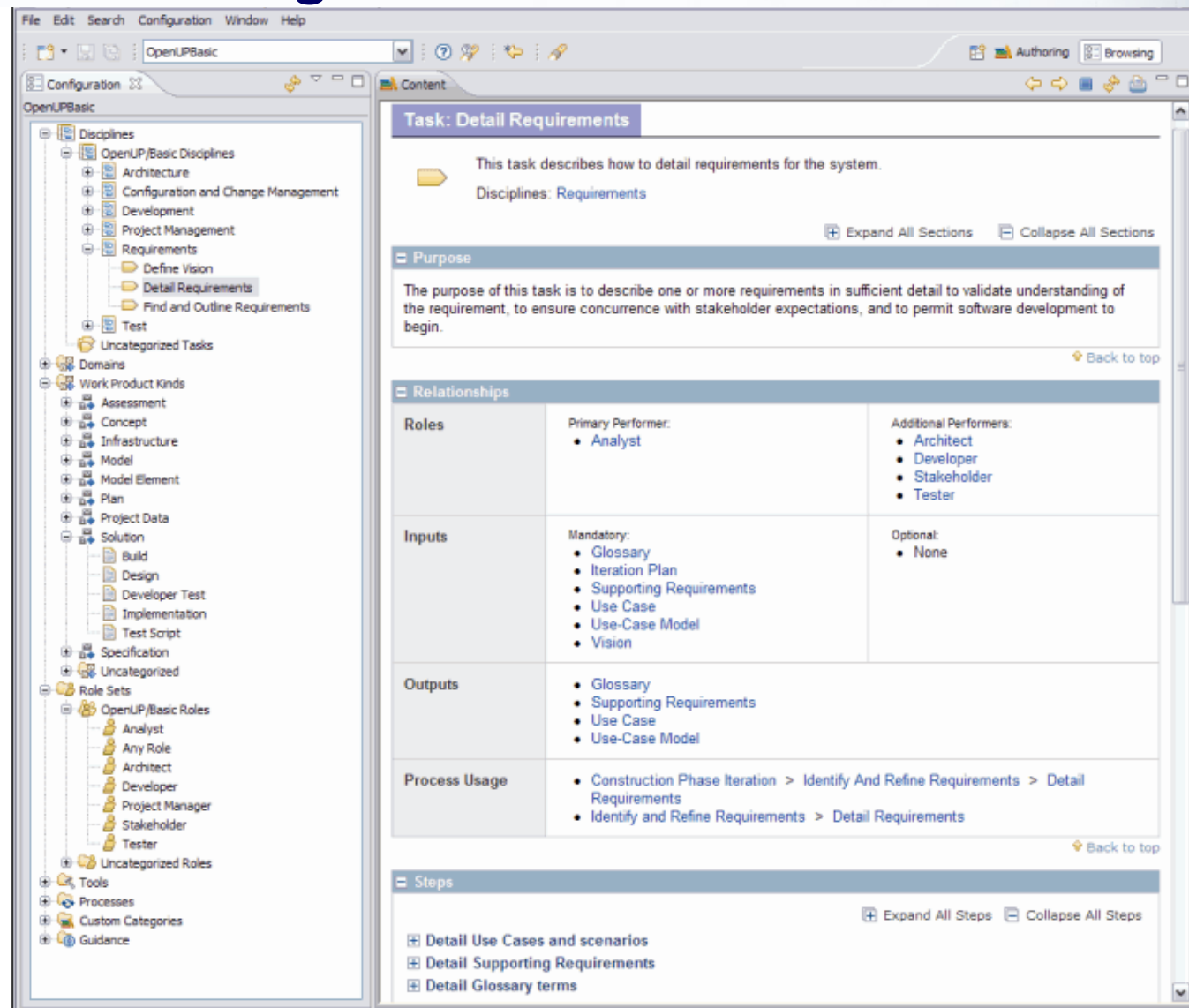
Books / Publications



Standards / Regulations



Homegrown Methods



The screenshot displays the Eclipse Process Framework Composer (EPF) interface. The left pane shows a hierarchical tree of tasks and roles. The right pane displays the details for the 'Task: Detail Requirements' task.

**Task: Detail Requirements**

This task describes how to detail requirements for the system.

Disciplines: Requirements

**Purpose**

The purpose of this task is to describe one or more requirements in sufficient detail to validate understanding of the requirement, to ensure concurrence with stakeholder expectations, and to permit software development to begin.

**Relationships**

Roles	Primary Performer:	Additional Performers:
	Analyst	<ul style="list-style-type: none"> <li>Architect</li> <li>Developer</li> <li>Stakeholder</li> <li>Tester</li> </ul>

Inputs	Mandatory:	Optional:
	<ul style="list-style-type: none"> <li>Glossary</li> <li>Iteration Plan</li> <li>Supporting Requirements</li> <li>Use Case</li> <li>Use-Case Model</li> <li>Vision</li> </ul>	<ul style="list-style-type: none"> <li>None</li> </ul>

Outputs	
	<ul style="list-style-type: none"> <li>Glossary</li> <li>Supporting Requirements</li> <li>Use Case</li> <li>Use-Case Model</li> </ul>

**Process Usage**

- Construction Phase Iteration > Identify And Refine Requirements > Detail Requirements
- Identify and Refine Requirements > Detail Requirements

**Steps**

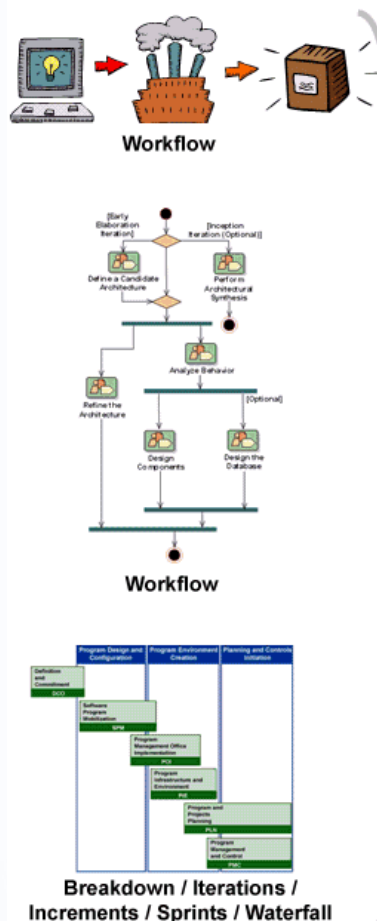
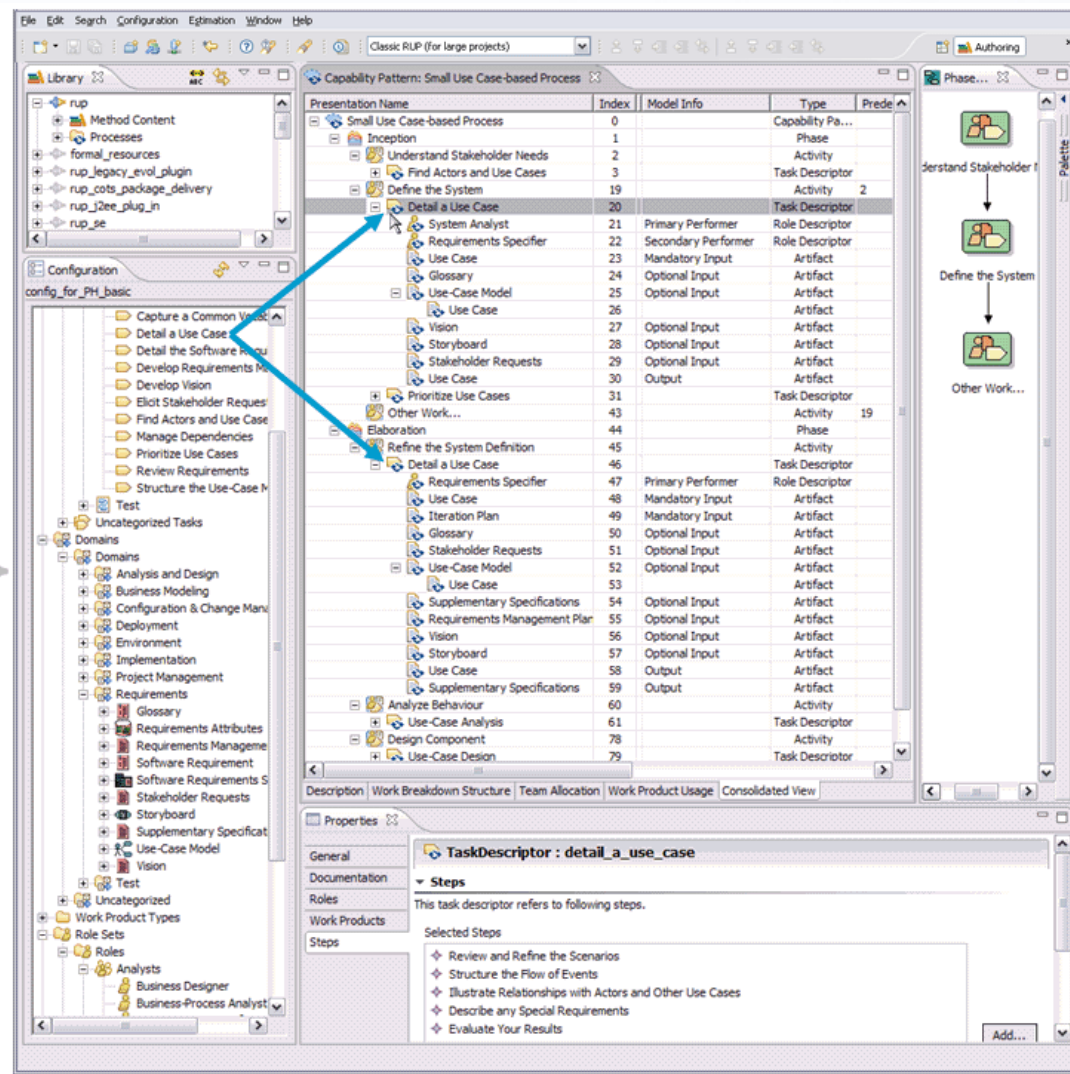
- Detail Use Cases and scenarios
- Detail Supporting Requirements
- Detail Glossary terms

Haumer, P. 2007. Eclipse Process Framework Composer Part 1 Key Concepts. <http://www.eclipse.org/epf/general/EPFComposerOverviewPart1.pdf>.

Presentation for the INCOSE Symposium 2011 Denver, CO USA



# Aladin : Process authoring

Presentation Name	Index	Model Info	Type	Predecessor
Small Use Case-based Process	0		Capability Pa...	
Inception	1		Phase	
Understand Stakeholder Needs	2		Activity	
Find Actors and Use Cases	3		Task Descriptor	
Define the System	19		Activity	2
Detail a Use Case	20		Task Descriptor	
System Analyst	21	Primary Performer	Role Descriptor	
Requirements Specifier	22	Secondary Performer	Role Descriptor	
Use Case	23	Mandatory Input	Artifact	
Glossary	24	Optional Input	Artifact	
Use-Case Model	25	Optional Input	Artifact	
Use Case	26		Artifact	
Vision	27	Optional Input	Artifact	
Storyboard	28	Optional Input	Artifact	
Stakeholder Requests	29	Optional Input	Artifact	
Use Case	30	Output	Artifact	
Prioritize Use Cases	31		Task Descriptor	
Other Work...	43		Activity	19
Elaboration	44		Phase	
Refine the System Definition	45		Activity	
Detail a Use Case	46		Task Descriptor	
Requirements Specifier	47	Primary Performer	Role Descriptor	
Use Case	48	Mandatory Input	Artifact	
Iteration Plan	49	Mandatory Input	Artifact	
Glossary	50	Optional Input	Artifact	
Stakeholder Requests	51	Optional Input	Artifact	
Use-Case Model	52	Optional Input	Artifact	
Use Case	53		Artifact	
Supplementary Specifications	54	Optional Input	Artifact	
Requirements Management Plan	55	Optional Input	Artifact	
Vision	56	Optional Input	Artifact	
Storyboard	57	Optional Input	Artifact	
Use Case	58	Output	Artifact	
Supplementary Specifications	59	Output	Artifact	
Analyze Behaviour	60		Activity	
Use-Case Analysis	61		Task Descriptor	
Design Component	78		Activity	
Use-Case Design	79		Task Descriptor	

Haumer, P. 2007. Eclipse Process Framework Composer Part 1 Key Concepts. <http://www.eclipse.org/epf/general/EPFComposerOverviewPart1.pdf>.

Presentation for the INCOSE Symposium 2011 Denver, CO USA

# Aladin Process Patterns Support (1)

Process Pattern 01

## Process Pattern: Process Pattern 01

► General Information

▼ Context Information

Provide context information about this process pattern.

Problem Description:

Affected Disciplines:

☐ Manage Engineering Effort
 ☐ Analyze Capability Gap
 ☐ Develop Operational Architecture
 ☐ Develop SoS Architecture
 ☐ Assess Force Development Options
 ☐ All

Affected Stages and Iterations:

☐ Inception
 ☐ Comprehension Iteration 1
 ☐ Comprehension Iteration 2
 ☐ Elaboration Iteration 1
 ☐ Elaboration Iteration 2
 ☐ Elaboration Iteration 3
 ☐ Completion
 ☐ All

Team Characteristics:

Size: 
 Distributed: 
 Cohesion: 
 CEP knowledge: 
 Gap knowledge:

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

# Aladin Process Patterns Support (2)

**Search**

Method Search | **Process Pattern Search**

Search ID :

**Capability Engineering Initiative**

Gap Complexity:  ☐ As-is before To-be

Gap Novelty:  ☐ Ongoing Initiative

**Stages**

☐ Inception ☐ Comprehens ☐ Elaboration ☐ Completion

**Project Surrounding**

Timeframe:  Classified:

Budget:  IT Support:

**Deliverables**

☐ Force Development Options ☐ SoS Architecture

☐ Operational Architecture ☐ Requirements

☐ Strategic Context Analysis ☐ Engineering Management Plan

Number of corresponding process patterns: 5

**Search**

Method Search | **Process Pattern Search**

Search ID :

**Team Characteristics**

Size:  Cohesion:

CEP Knowledge:  Gap Knowledge:

Distributed:

**Roles Staffing**

CET Lead:  CET Coord.:

FDO Analyst:  Arch. Modeler:

OR Analyst:  SOS Architect:

IT Specialist:  Op. Architect:

Req. Analyst:

Number of corresponding process patterns: 5

## Some Process patterns :

- “Lite” family or patterns (a pattern per discipline)
  - Each discipline workflow is reduced to its critical activities.
- “Reuse” family (a pattern per discipline)
  - Deliverables & workproducts from a previous similar project are reused
- “High Spin”
  - No stages, but quick “sprints”, to build up quickly the solution through fast-paced iterations.
- “One Option” family of patterns
  - One part of the solution is “given” by the capability manager.
- “As-is first”
  - “As-Is” option is delivered first, instead of being concurrently developed with the others.
- “Early Close-out”
  - Premature close-out due to external events.



# High Spin – Forces in Presence

## For

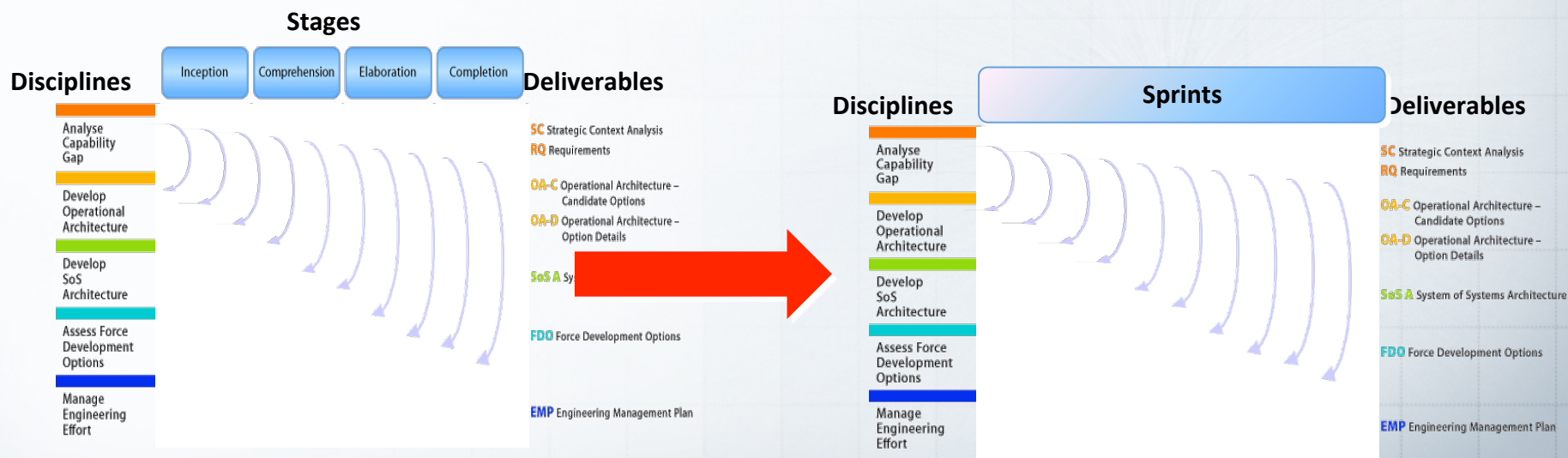
- Ill-defined gap
- Continuously evolving requirements
- Time-boxed initiative
- Client rep. on site

## Against

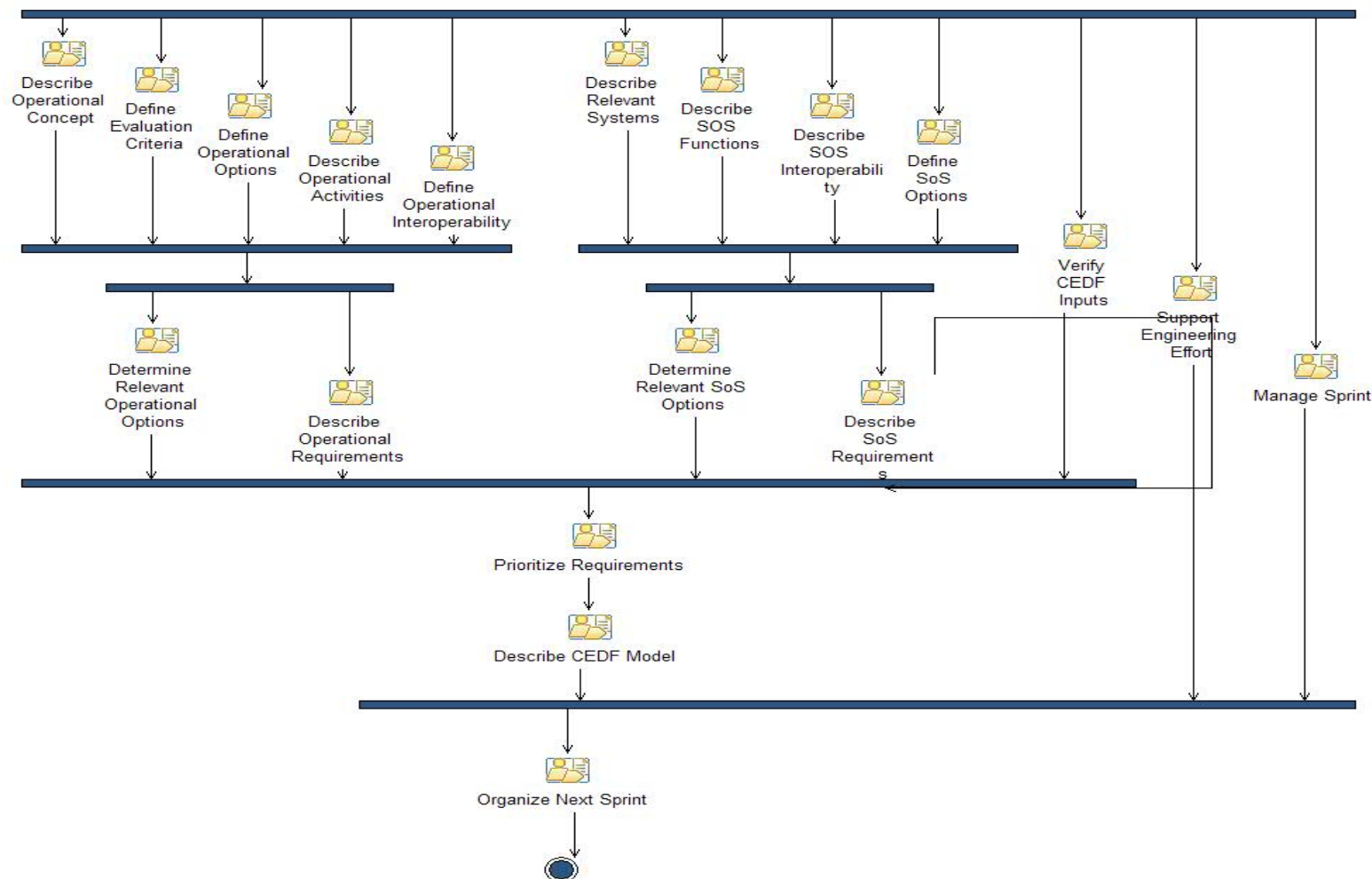
- Low Stakeholders availability
- Inexperienced CET
- Low collaboration
- Info & knowledge hard to find/use

# High-spin – Solution Overview

- All stages merged into a “sprint”
  - Effort dispensed in short bursts
  - Option refined iteratively until client agreement
- Process self evolving according to knowledge discovery



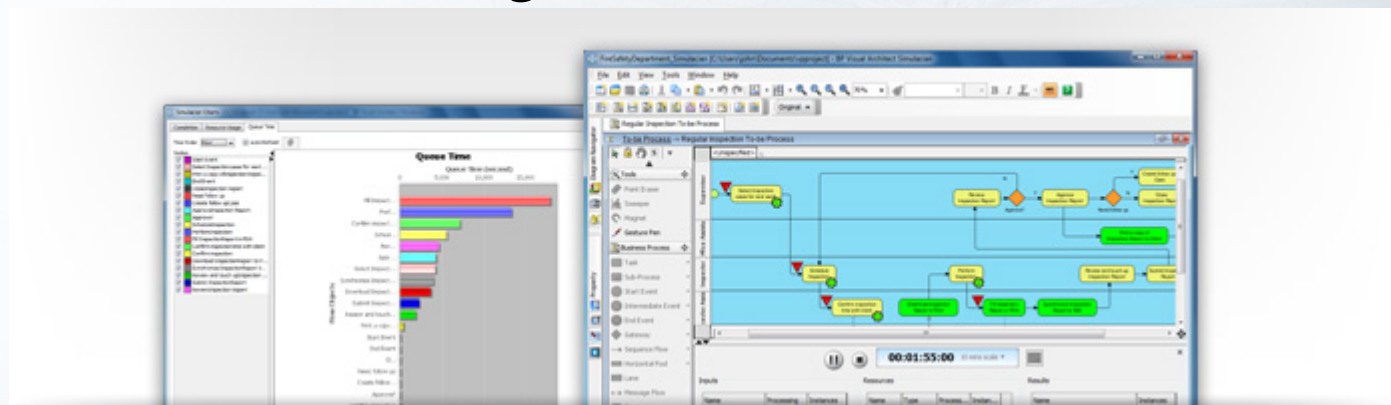
# High Spin - Solution



## How to measure process performance ?

- Different methods reviewed
  - Comprehensive audit
  - Performance dashboard
  - Capability Maturity Model
  - University of California Approach
  - Earned-Value Report
  - INCOSE Method
  - A mix of the above

**All of the above imply real experimentation.**  
***Is there a way to predict performance improvement through simulation ?***



Visual-paradigm simulation illustration

<http://www.visual-paradigm.com/product/bpva/provides/simulacion.jsp>



# Experimentation versus Simulation

## Experimentation

- Repeatability & Reproducibility
  - Medium to large schedule projects
  - Project specificity
  - Cost \$\$\$
- Learning curve bias influences results
- Provide real-world results

## Simulation

- Eliminates constraints
  - Time, budget, resources availability
- Allows repeatability
- Allows “what-if” analysis
- Eliminate cognitive and learning curve biases
- What is the validity of results?

## Simulation (Way ahead)

- We are looking for ways to assess processes through simulation,
  - Through Earned-Value Reports .
  - Using agent simulation tools:
    - NetLogo,
      - multi agent programmable environment.
      - North-western University, Center for Connected Learning and Computer-Based Modeling.
    - Kepler(Kepler-project.org),
      - scientific workflow application.
      - UC Davis, UC Santa Barbara, and UC San Diego.
  - Using Business Process simulation software
    - Visual-Paradigm, YAWL, Global360 ...

# Conclusion and way ahead

- We have gone through
  - Software engineering concepts applied to capability engineering;
  - Process patterns for improving process agility
    - Identification of High-level process patterns;
  - Agility improvement assessment
    - Through experimentation or simulation ;
- Simulation tools under investigation / development.
- Lower-level patterns to be identified too.

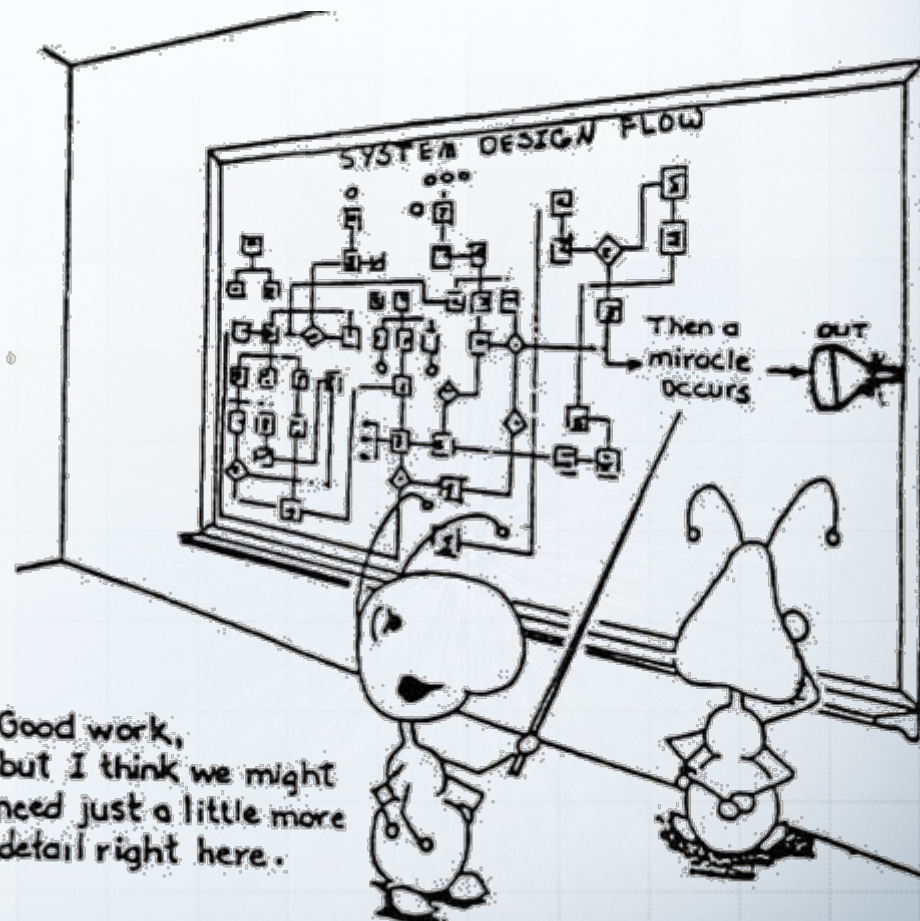
# Process Engineering, a research theme

- Excerpts from the “Process Research Framework” :
- How to define and build processes and understand their performance?
  - ✓ E3 Providing process engineering infrastructure
    - Determine infrastructure needed to best support this process engineering environment
  - ✓ E2 Organizing processes for reuse
    - Engineer, assemble, combine and reuse process components to meet performance targets
  - E1 Specifying processes using evidence
    - Specify processes with adequate empirical evidence of their performance





# Comments, Questions ....



For more information :

Christophe Necaille

[Christophe.necaille@drdc-rddc.gc.ca](mailto:Christophe.necaille@drdc-rddc.gc.ca)

<http://www.drdc-rddc.gc.ca>

<http://benwarsop.files.wordpress.com/2010/06/then-a-miracle-occurs.jpg>

