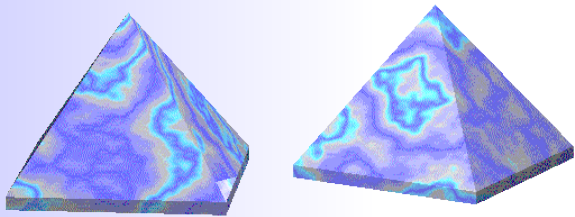


# UML and the Unified Process

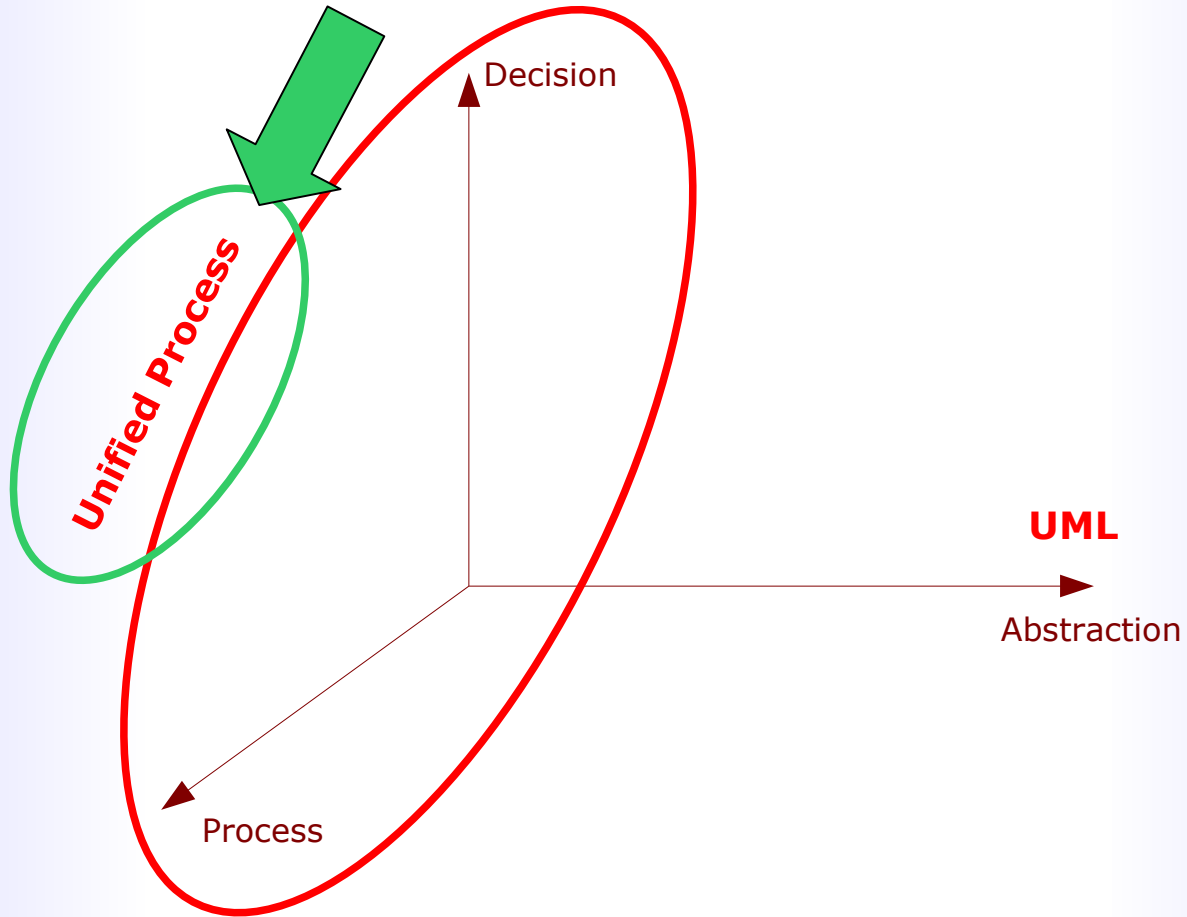


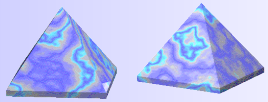
**FloConsult SPRL**

<http://www.floconsult.be>  
<mailto:info@floconsult.be>

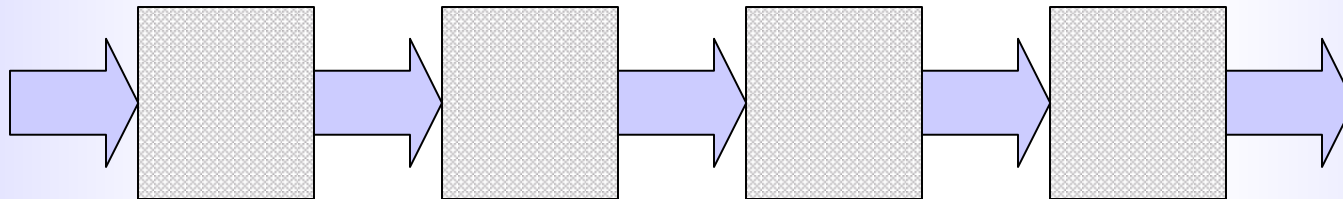
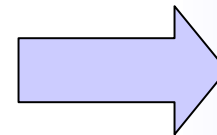
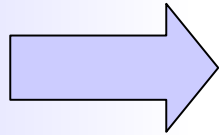
**Renaud FLORQUIN**  
**Isabelle LECLERCQ**

# !! OO Process ≠ UML

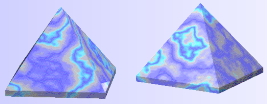




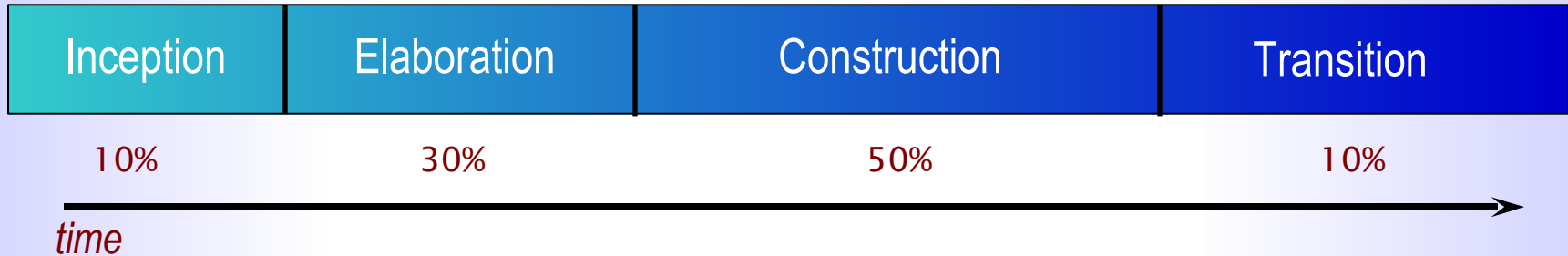
# OO SW development (1)



- Iterative/incremental
- Risk-driven
- Architecture driven



# Unified process phases



## ● Inception

- ◆ Define the scope of the project and develop business case

## ● Elaboration

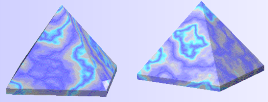
- ◆ Plan project, specify features and baseline the architecture

## ● Construction

- ◆ Build the product

## ● Transition

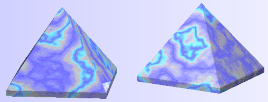
- ◆ Transition the product to users



# UP practices

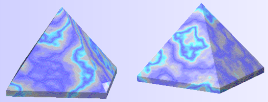


1. Iterative/incremental software development approach
2. Risk management
3. Use Case driven
4. Architecture Centric
5. Supported by UML



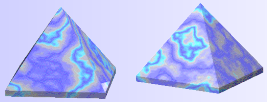
# 1. Iterative/Incremental

- Iterative development
  - ◆ Rework scheduling strategy to revise and improve parts of the system
  - ◆ Refactoring
    - e.g. only 10 % of the code should be optimized for performance
- Incremental development
  - ◆ The software is developed and delivered in stages (increments)

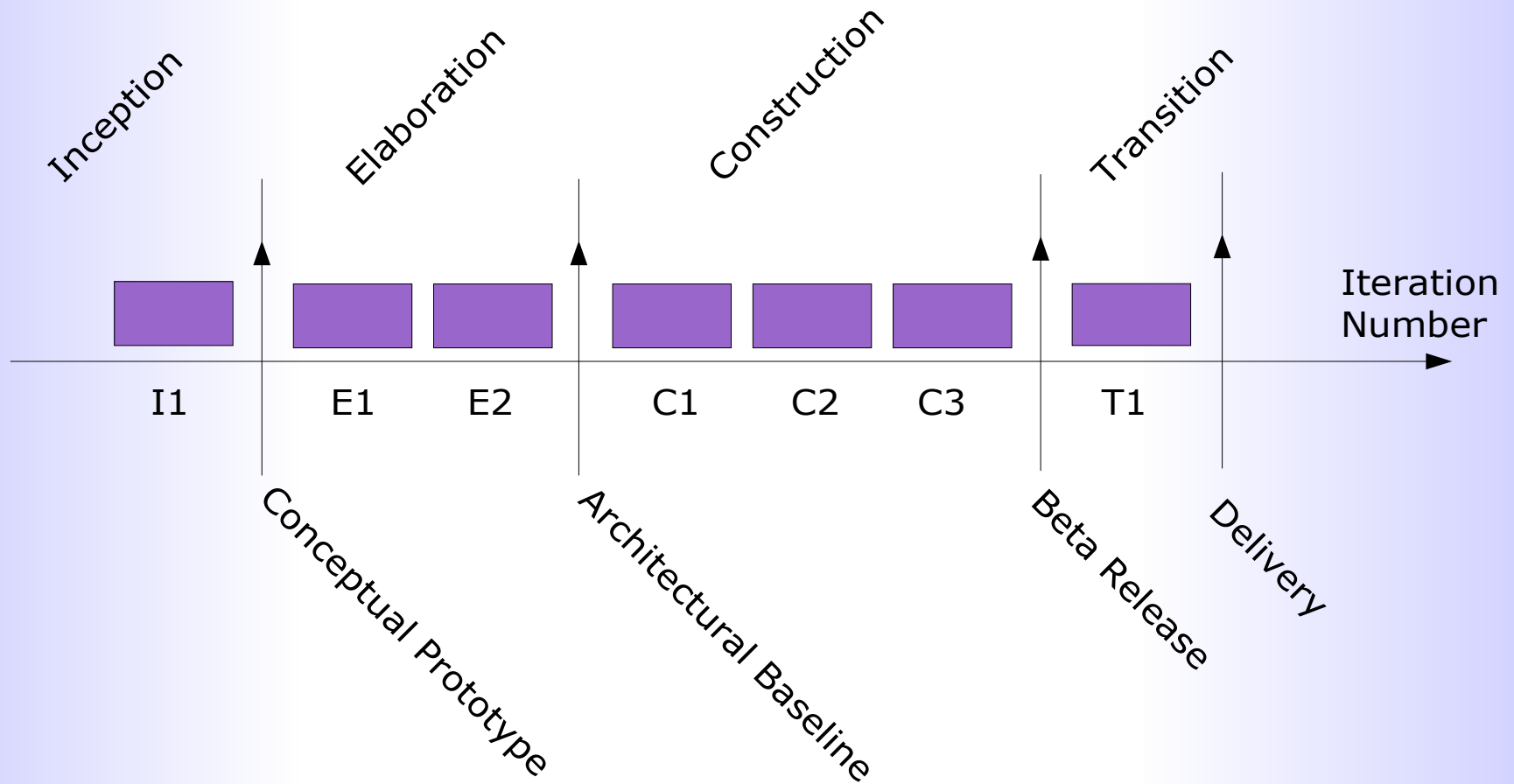


# UP: iterations

- Number of iterations
  - ◆ Depends on the process phase
  - ◆ Generally ranges from 0 to 3
- Duration of iterations
  - ◆ Depends on the size of the project
  - ◆ From 2 weeks to 6 months
- Time-boxing:
  - ◆ Iterations are **fixed** in length
  - ◆ If needed, reduce scope, rather than slip completion date

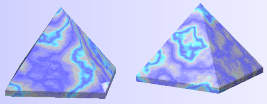


# UP iterations: small project

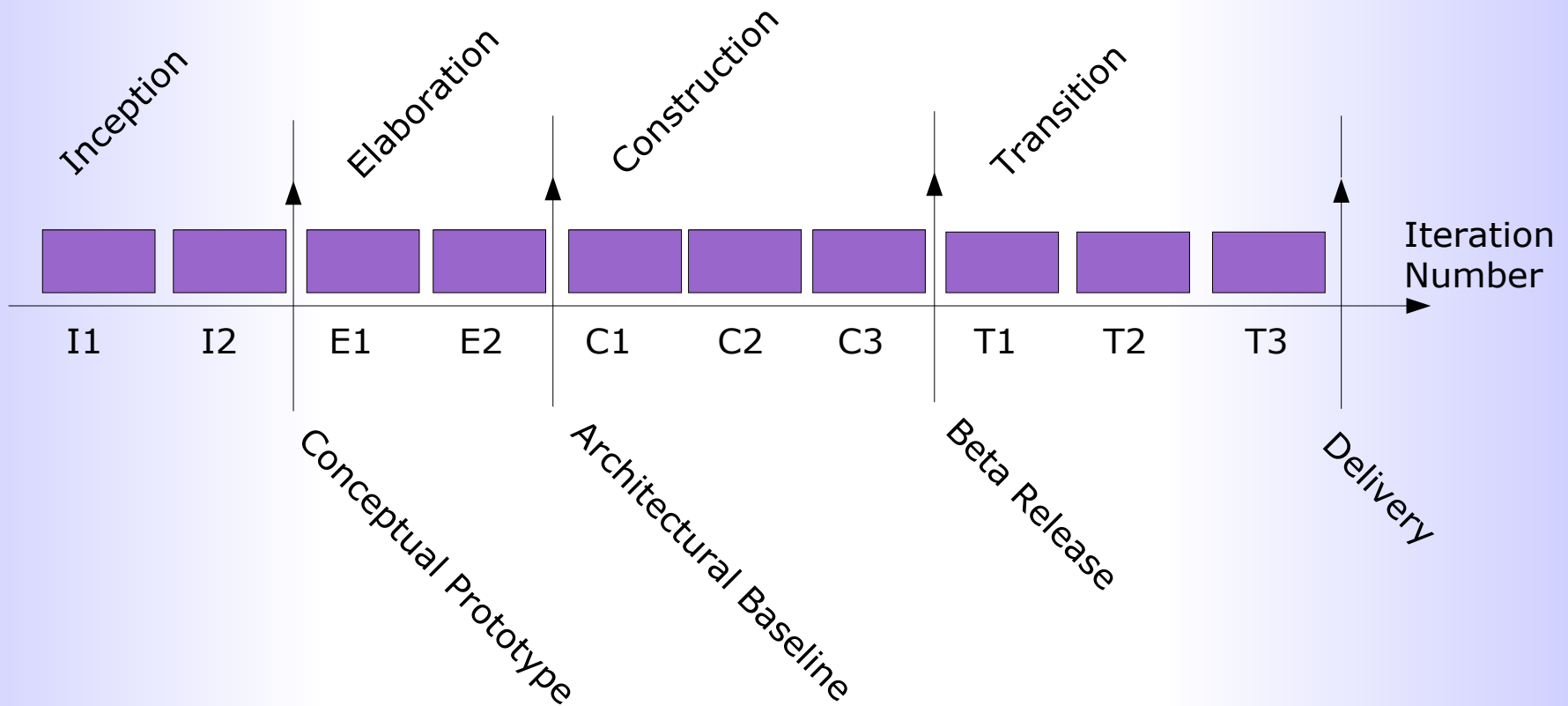


Iterations: from 2 to 4 weeks

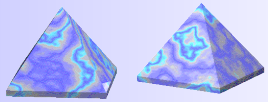




# UP iterations: larger project



Iterations: from 4 to 24 weeks



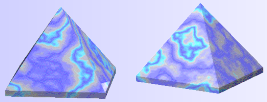
## 2. Risk management (1)

### ● Risk reduction

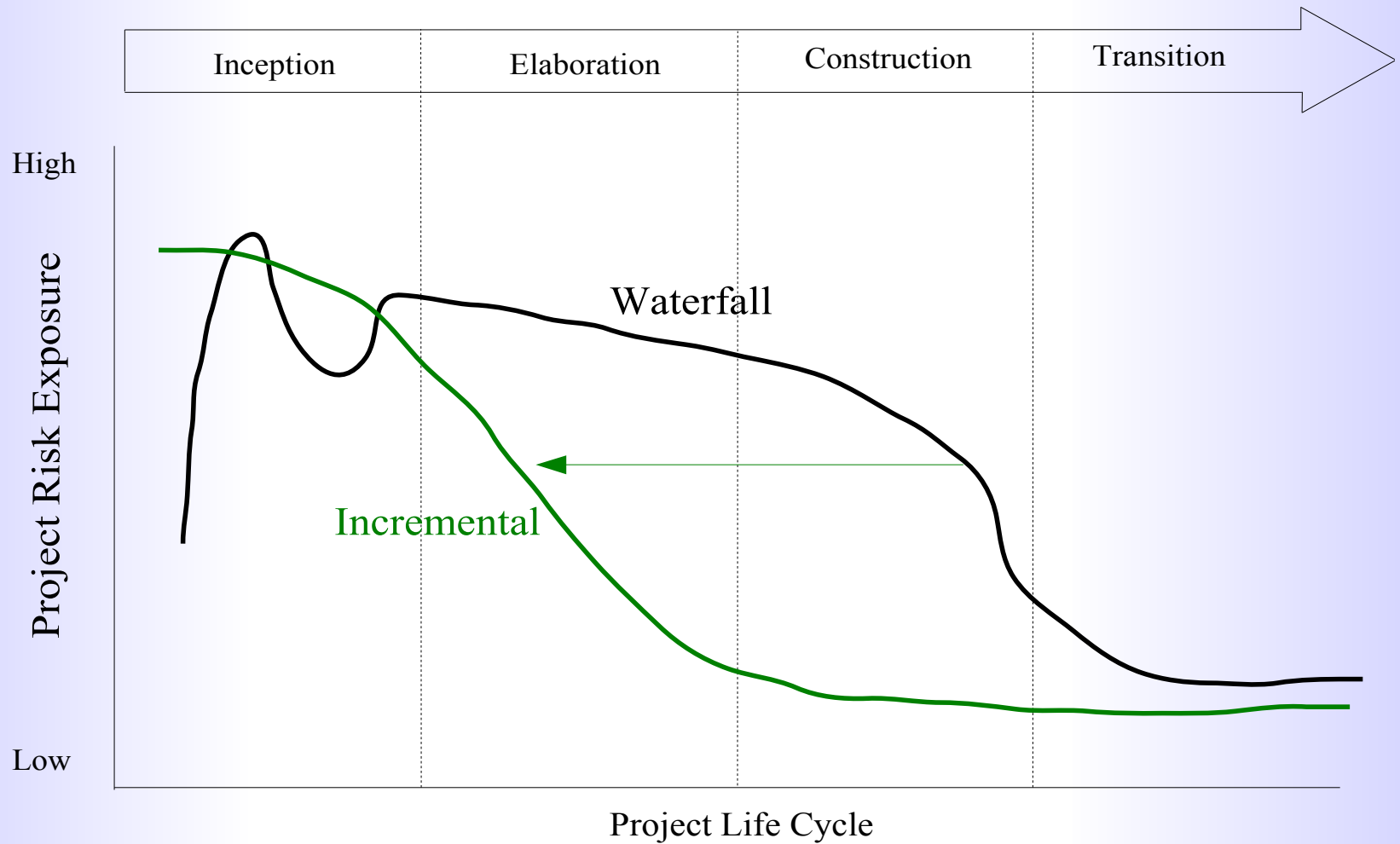
- ◆ Breaks the project into small sub-projects
- ◆ Possibility to handle riskiest sub-projects at first

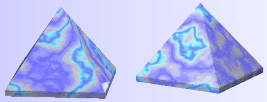
### ● Elaboration (early):

- ◆ Understand better the requirements
- ◆ Baseline the architecture
  - "executable" architecture
  - Main components
  - Selection of technologies
  - ...



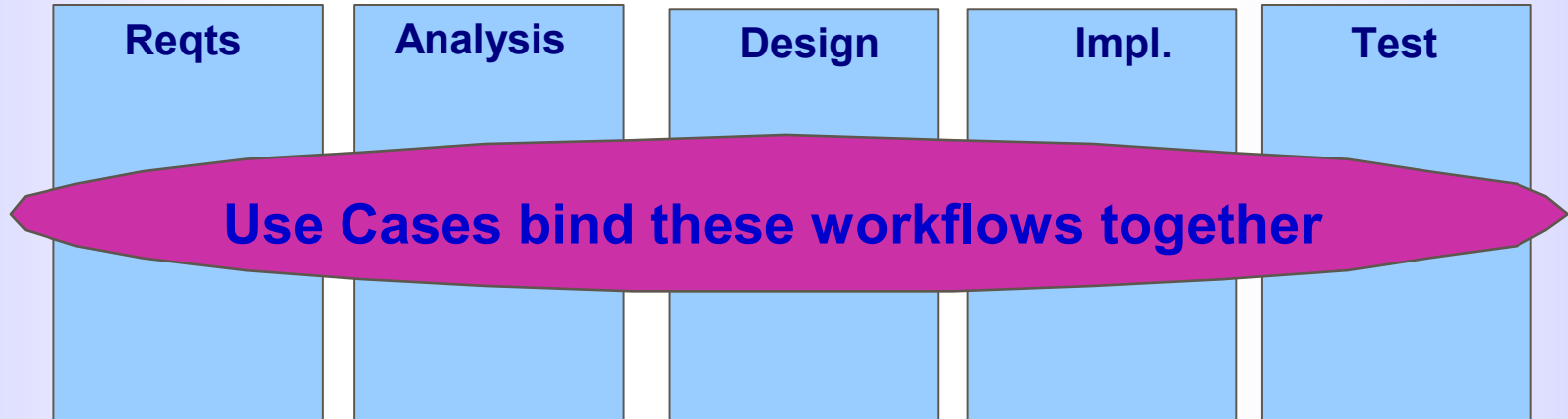
# Risks management (2)

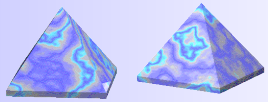




## 3. Use-case driven

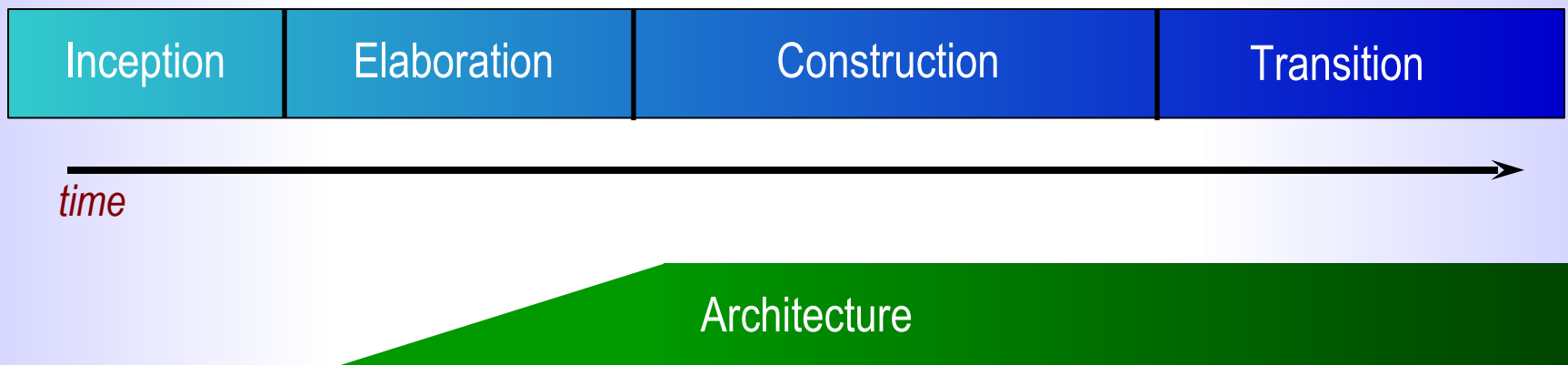
- **Basis for the entire development process**
  - ◆ Help to synchronize the content of various models
  - ◆ Drive numerous activities in the process
    - Capturing for functional requirements
    - Validation of the design model
    - Definition of test cases
    - Planning of iterations

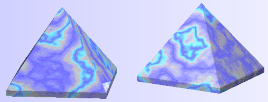




## 4. Architecture centric

- Models to visualize, specify, construct and document architecture
  - ◆ 4 + 1 (now N) views model of architecture
- Successive refinement of an executable architecture
  - ◆ Architectural prototypes for validation and baselines



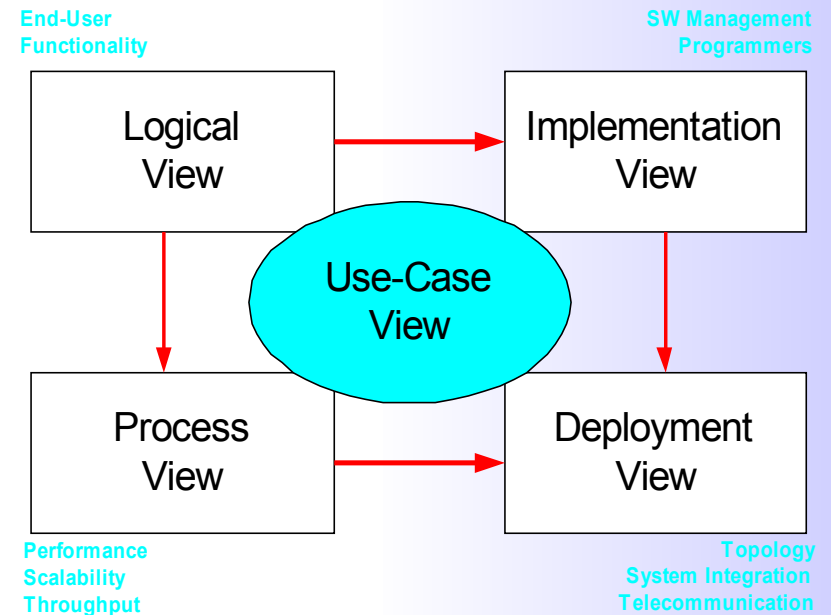


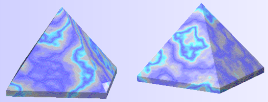
# 4+1 Views

## Views of a SW Architecture:

- ◆ Logical
- ◆ Process
- ◆ Implementation
- ◆ Deployment
- ◆ Use cases

## Now N+1 views

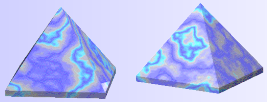




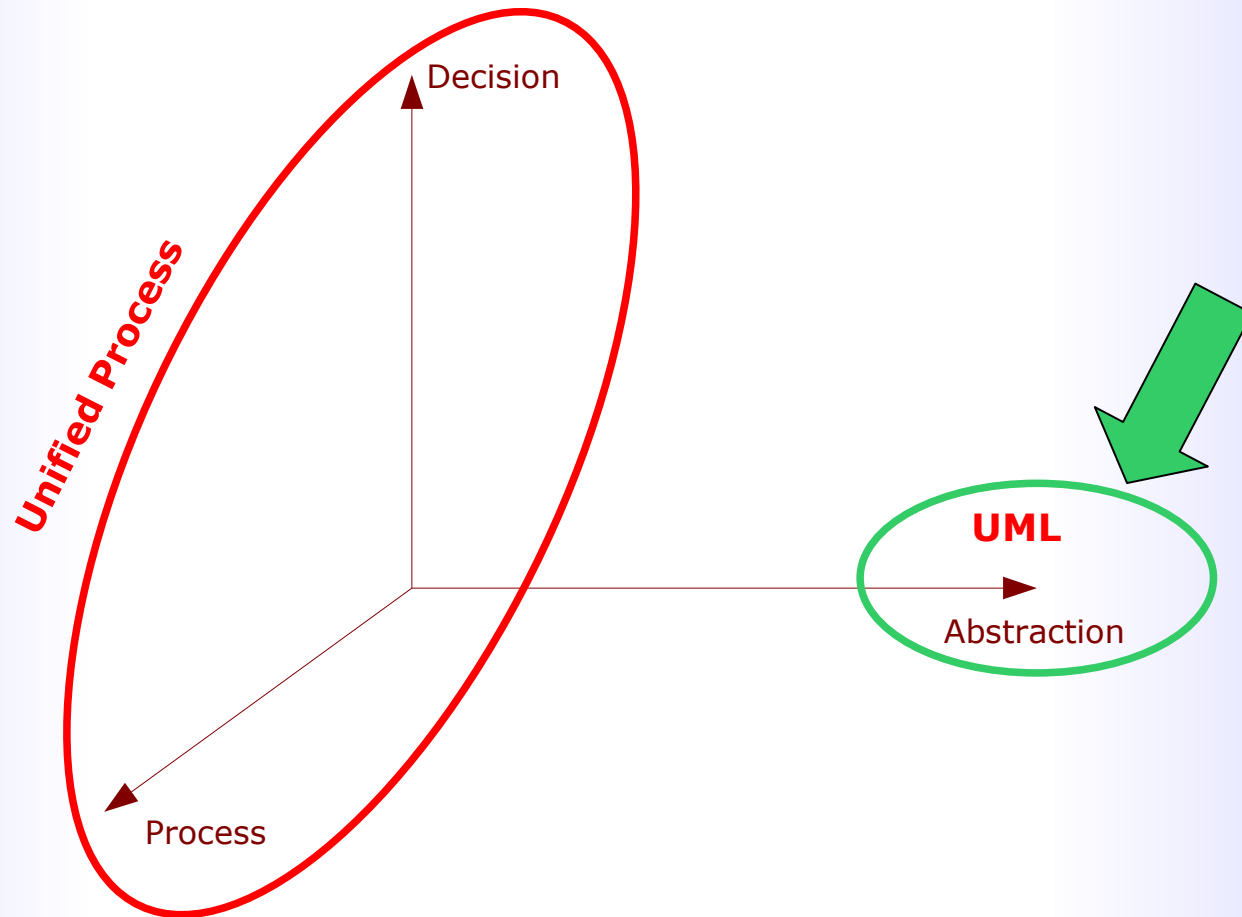
## 5. Supported by UML

- Visual modeling
- 13 diagrams
- Most of them Object-Oriented
  - Not Use Case
  - Activity, State, ... => depends !

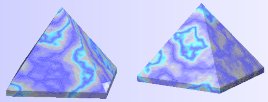




# !! OO Process $\neq$ UML



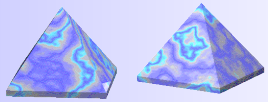




# BUT... What is UML ?



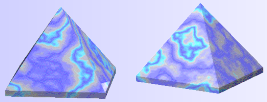
- Unified Modeling **Language**
- The UML is a graphical language for
  - ◆ Specifying
  - ◆ Visualizing
  - ◆ Constructing
  - ◆ Documenting
- Standardized by the **OMG**
  - ◆ [www.omg.org](http://www.omg.org)



# UML = Notation

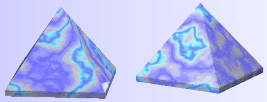
- UML: Takes the best of each of the 3 methods
  - ◆ OOSE (Jacobson) : Use Cases
  - ◆ OMT (Rumbaugh) : Analysis
  - ◆ Booch : Conception, Architecture
- And of many others:
  - ◆ Harel
  - ◆ Kruchten
- But only **notation**, impossible to standardize a unique process



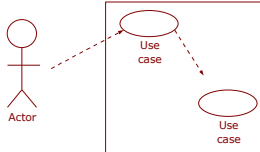
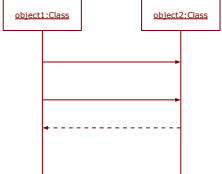
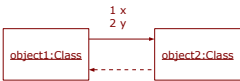
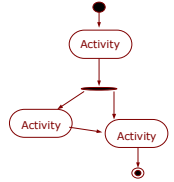
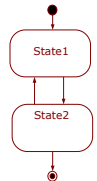


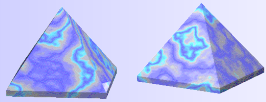
# UML static diagrams

Static Diagram	Notation	Used in ...	OO
<b>Class</b>		Inception <b>analysis/design</b> (Domain model)  Elaboration <b>analysis/design</b> (Design model)	✓
<b>Deployment</b>		Elaboration <b>analysis/design</b> (N Views)	
<b>Component</b>		Elaboration <b>analysis/design</b> (N Views)	✓

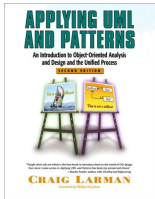
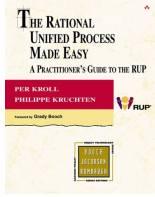
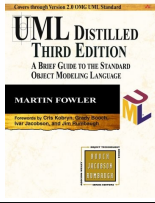


# UML dynamic diagrams

Dynamic Model	Notation	Used in ...	OO
<p><b>Use Case</b></p>		<p>Inception <b>requirements</b> (High-level, 20 % detailed)</p>	
<p><b>Sequence</b></p>		<p>Elaboration <b>requirements</b> (System Sequence Diagrams)</p> <p>Elaboration <b>analysis/design</b> (between objects)</p>	<p>(✓)</p>
<p><b>Collaboration (Communication)</b></p>		<p>Elaboration <b>analysis/design</b> (between objects)</p>	<p>✓</p>
<p><b>Activity</b></p>		<p>Elaboration <b>requirements</b> (to illustrate Use Cases)</p>	<p>(✓)</p>
<p><b>State</b></p>		<p>Elaboration <b>analysis/design</b> (between objects)</p>	<p>(✓)</p>



# Bibliography

	<p><b>Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development , 3rd Edition</b> by Craig Larman; 736 p, October 2004, Prentice Hall, ISBN 0-13-148906-2</p>
	<p><b>The Rational Unified Process made Easy: A Practitioner's guide to RUP</b> by Per Kroll, Philippe Kruchten; 464 p, April 2003, Addison-Wesley, ISBN 0-321-16609-4</p>
	<p><b>UML Distilled: A Brief Guide to the Standard Object Modeling Language, 3<sup>d</sup> Edition</b> by Martin Fowler; 192 p, September 2003, Addison-Wesley, ISBN 0-321-19368-7</p>