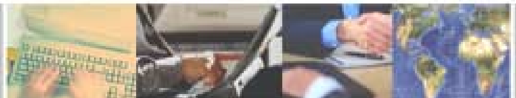
_experience the commitment

Model-Driven Architecture Case Study

June 12th, 2003

Presented by:
David Bertrand
Director-Consulting, CGI Group

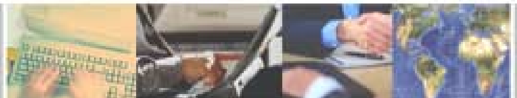
Table of content



Ingredients Of The Approach

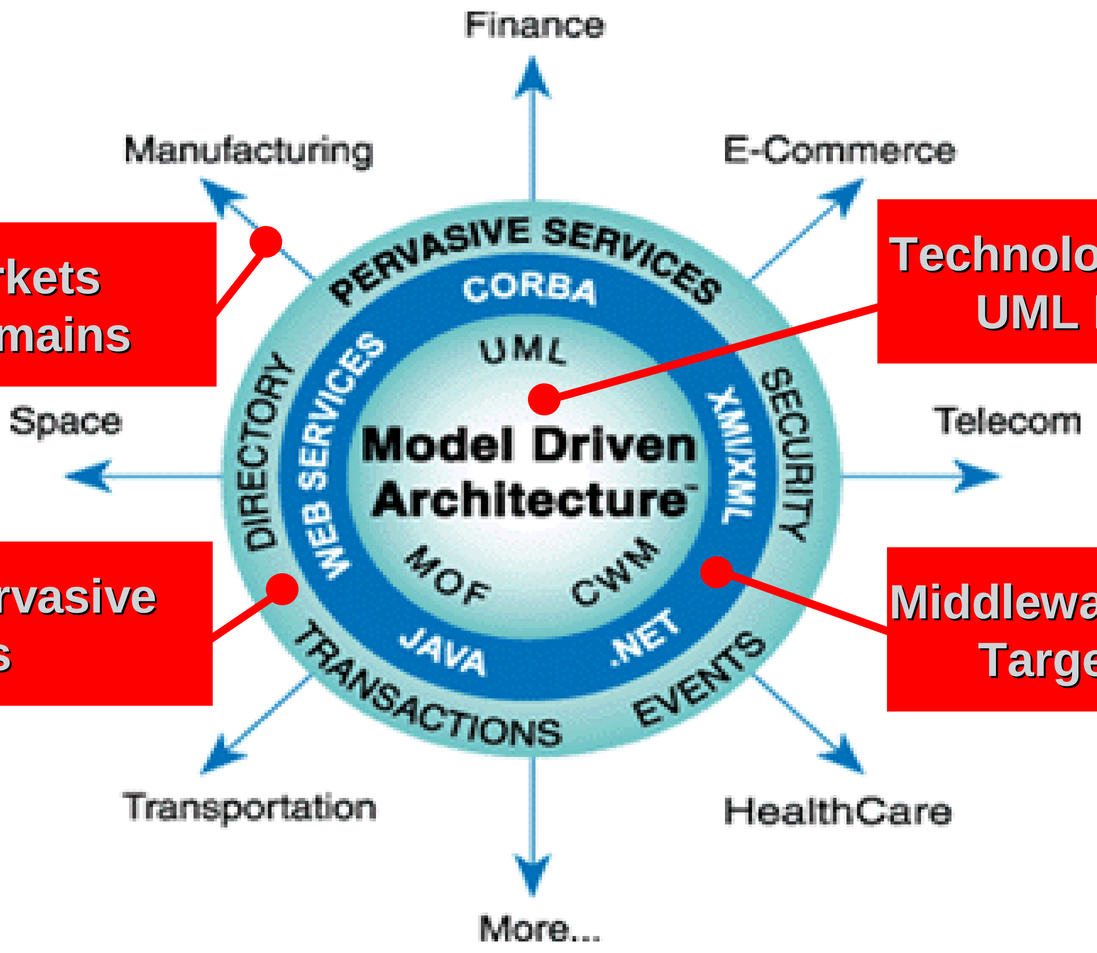
Main Ingredients

- Model-Driven Architecture (MDA)
- Rational Unified Process (RUP)
- Codagen Tools



Model-Driven Architecture

▪ MDA - Specifications



Vertical Markets Business Domains

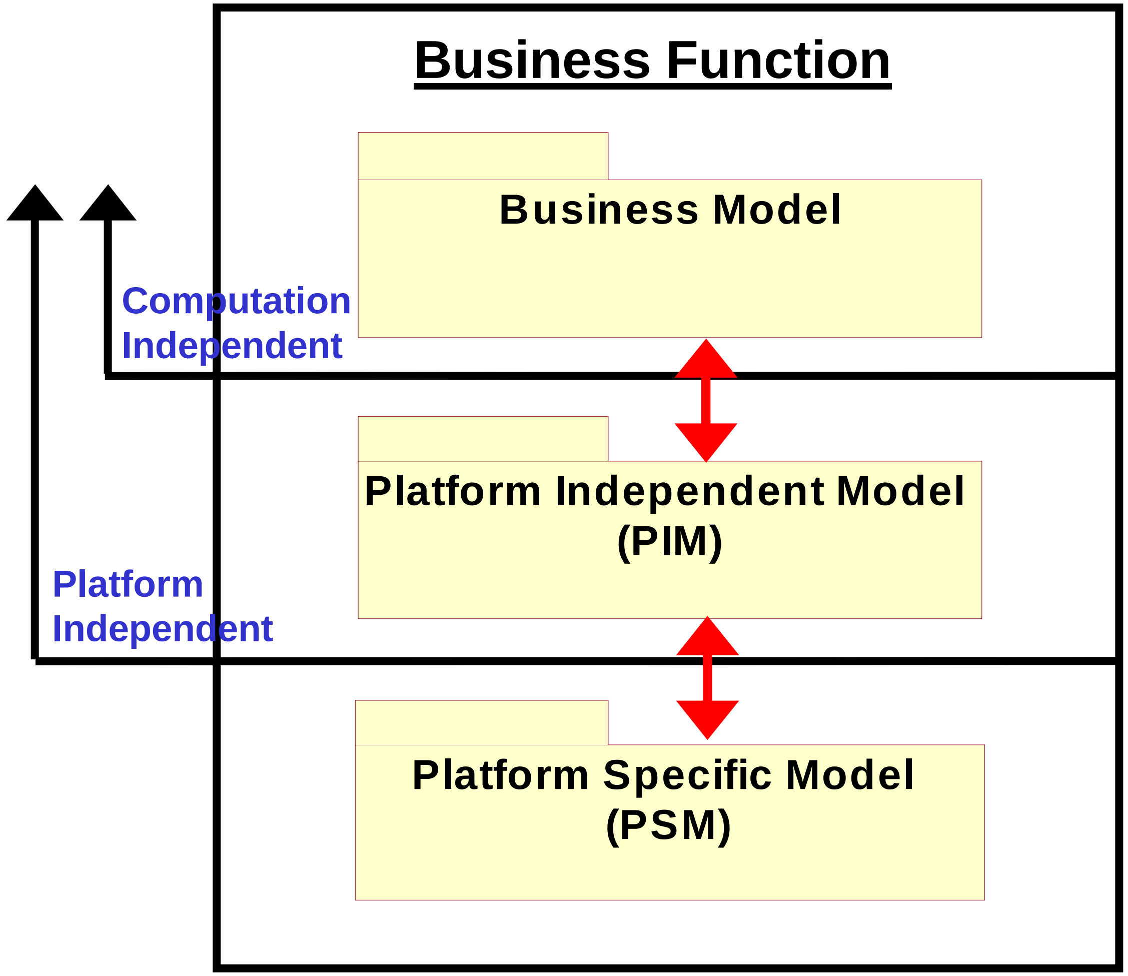
Technology-Independent UML Environment

Abstracted Pervasive Services

Middleware Environments Targeted For MDA

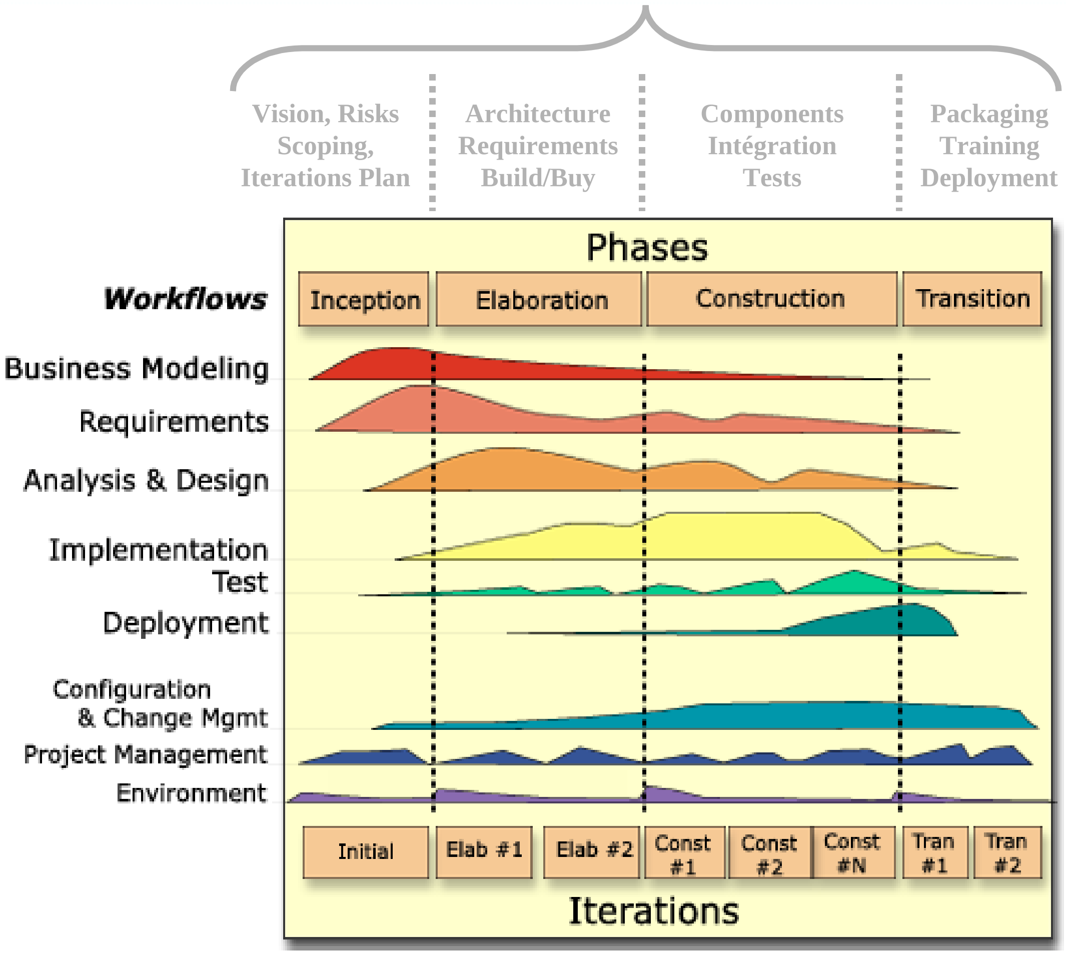
Model-Driven Architecture

- **MDA - Models**



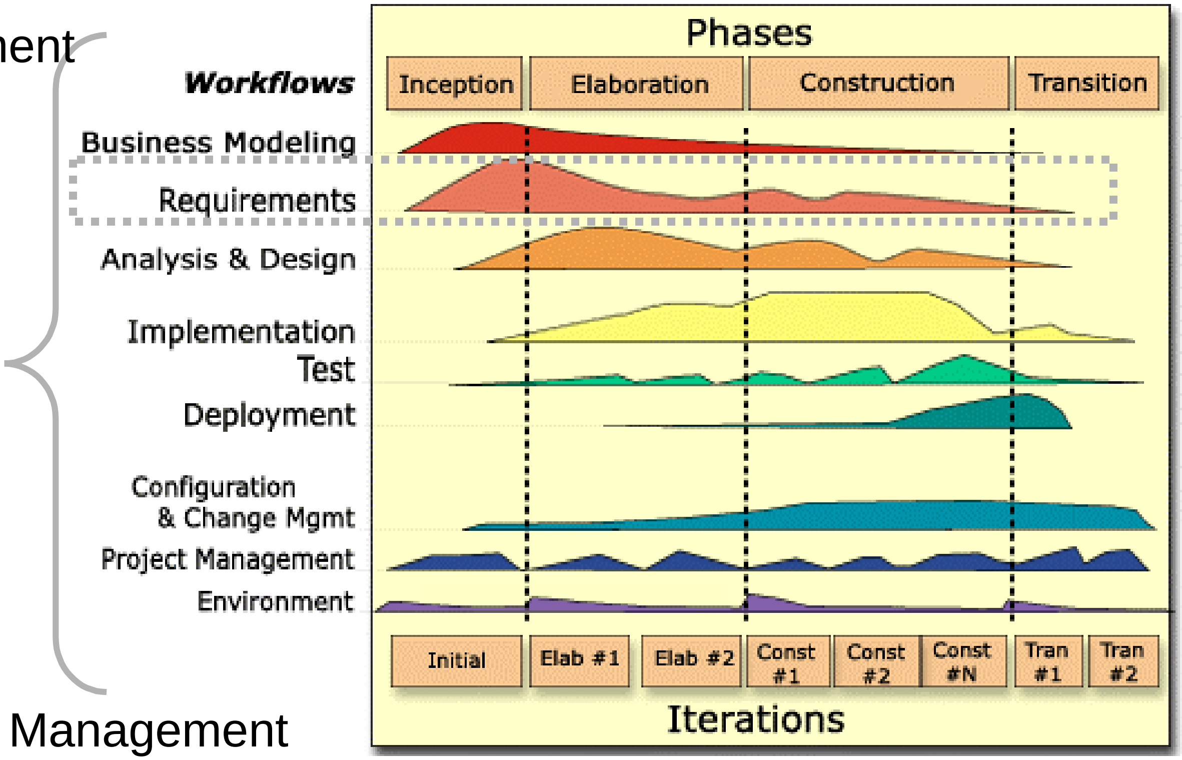
RUP Phases

- Inception
  - Vision
  - Risks
  - Scoping
  - Iterations Plan
- Elaboration
  - Requirements
  - Architecture
  - Built or Buy
- Construction
- Transition



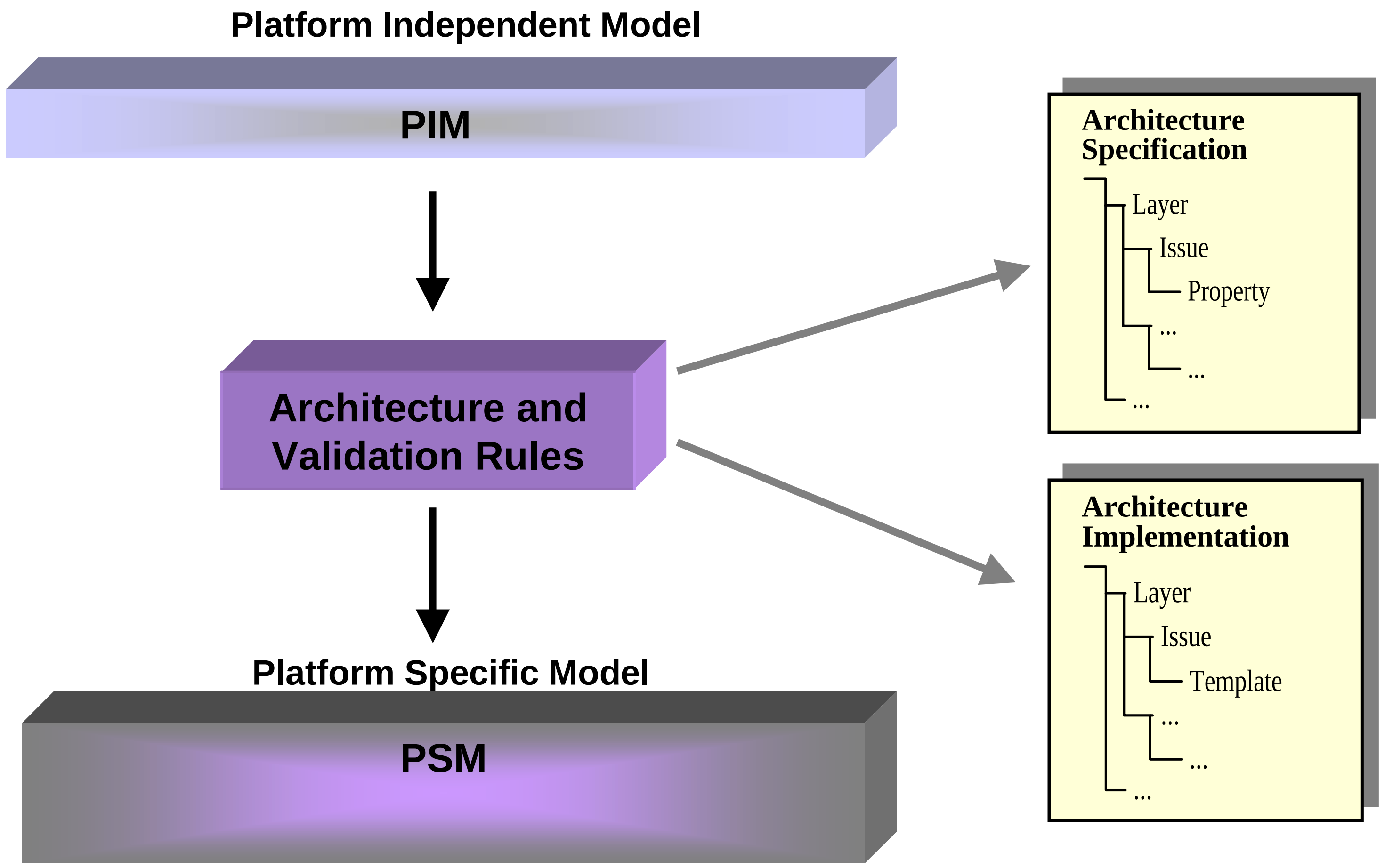
RUP Workflows

- Development Workflows
  - Requirements Management
  - Analysis & Design
  - Implementation
  - Tests
  - Deployment
- Support Workflows
  - Project Management
  - Configuration & Change Management
  - Tools & Environment



Codagen Tools

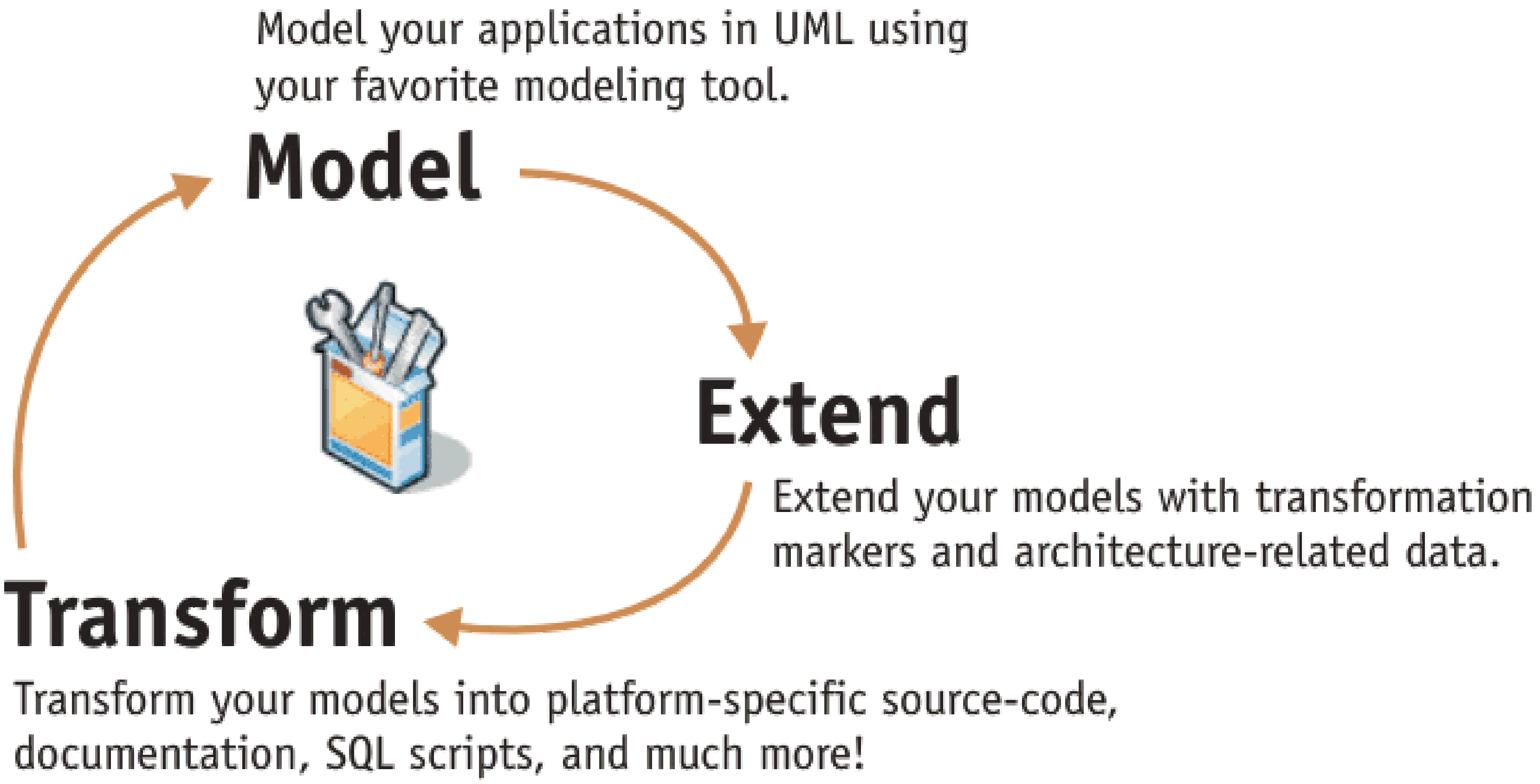
▪Model-Driven Architecture with Codagen



Platform Independent Model
PIM
Architecture and Validation Rules
Platform Specific Model
PSM
Architecture Specification
Layer
Issue
Property
...
...
...
Architecture Implementation
Layer
Issue
Template
...
...
...

Codagen Tools

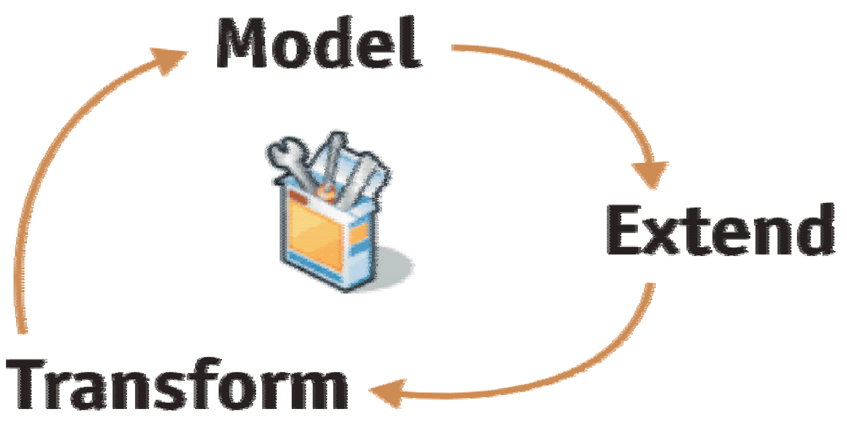
- **Model-Driven Architecture with Codagen**


Model your applications in UML using your favorite modeling tool.
Model
Extend
Extend your models with transformation markers and architecture-related data.
Transform
Transform your models into platform-specific source-code, documentation, SQL scripts, and much more!

Codagen Tools

- **Model-Driven Architecture with Codagen**

Direct support (add-in) for:
Rational Rose, TogetherSoft C/C, Microsoft Visio UML



Extend models by:
Defining and using UML Profiles

SPECIFY **MAP**

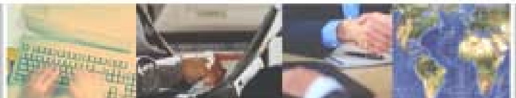
Transform models into:
Java, VisualBasic.NET, C#.NET, C++
ASCII files (XML, ASP, JSP, HTML, etc.)

IMPLEMENT **GENERATE**

Codagen Tools

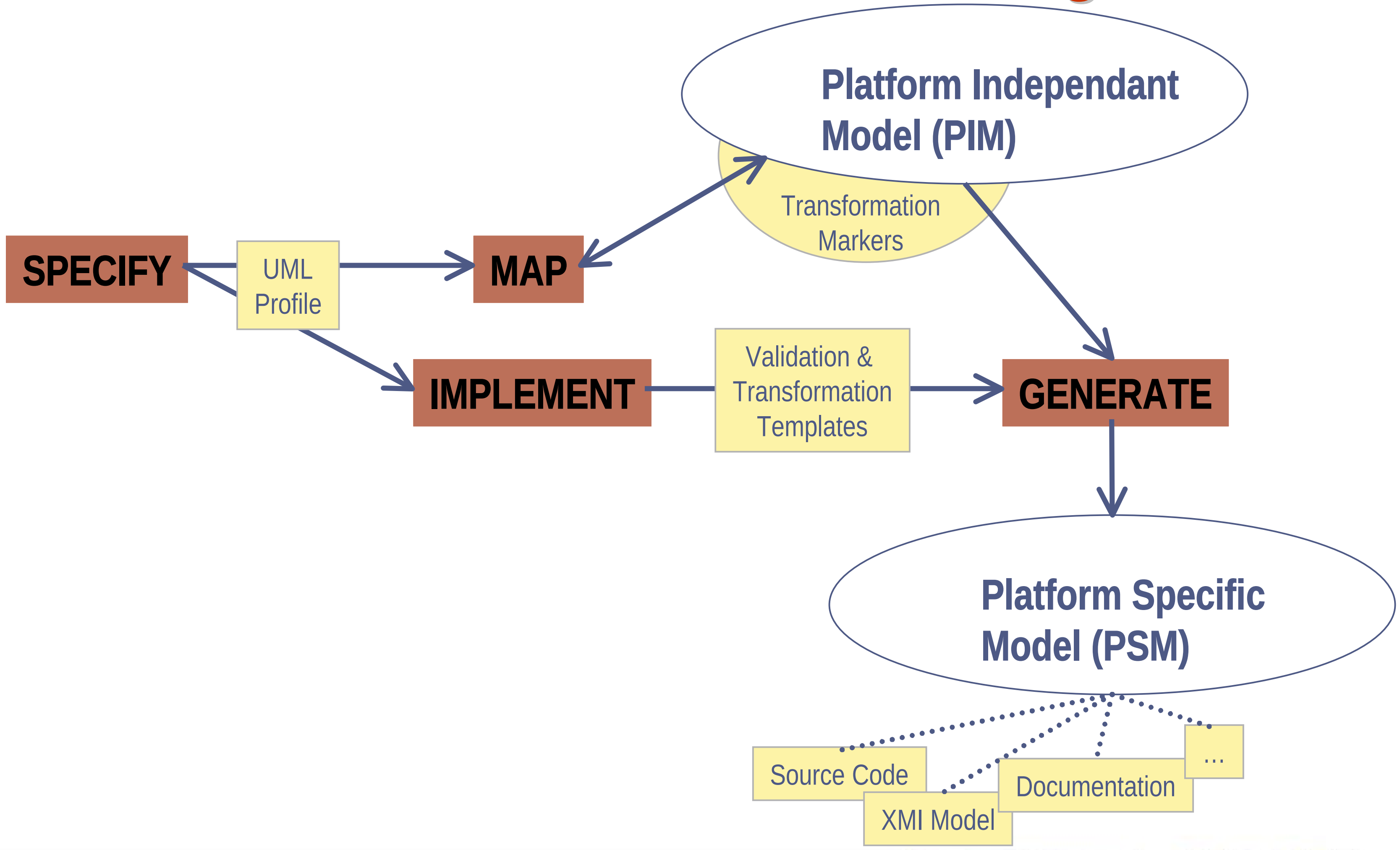
▪Model-Driven Architecture with Codagen

SPECIFY	Used to define a UML Profile which is a custom definition of tagged values that will help refine the model.
MAP	Used to map platform-specific data to UML elements such as Packages, Classes, Attributes, etc.
IMPLEMENT	Used to create and organize validation & transformation templates.
GENERATE	Used to generate the code based on the mapping and the templates.



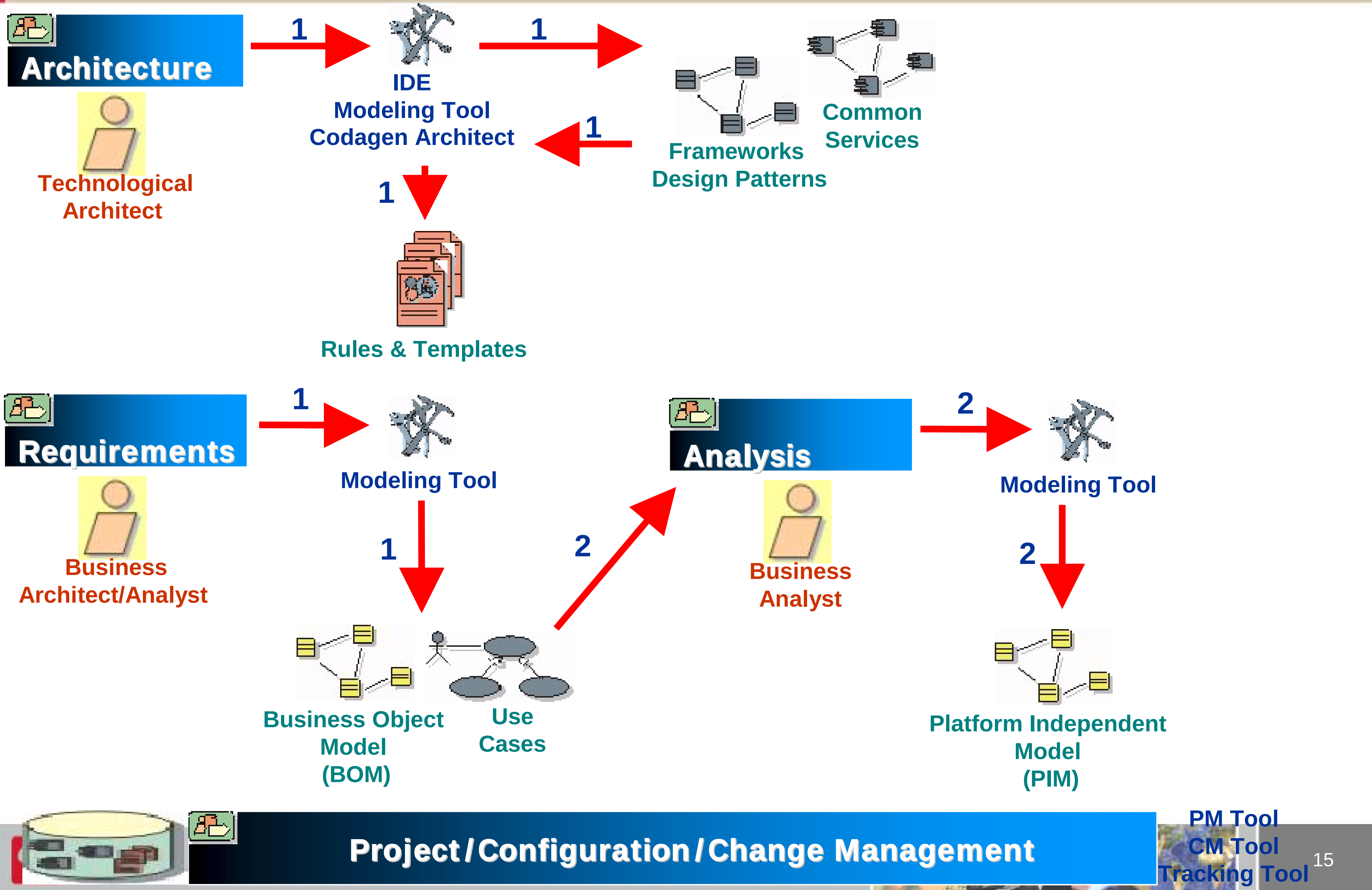
Codagen Tools

▪Model-Driven Architecture with Codagen

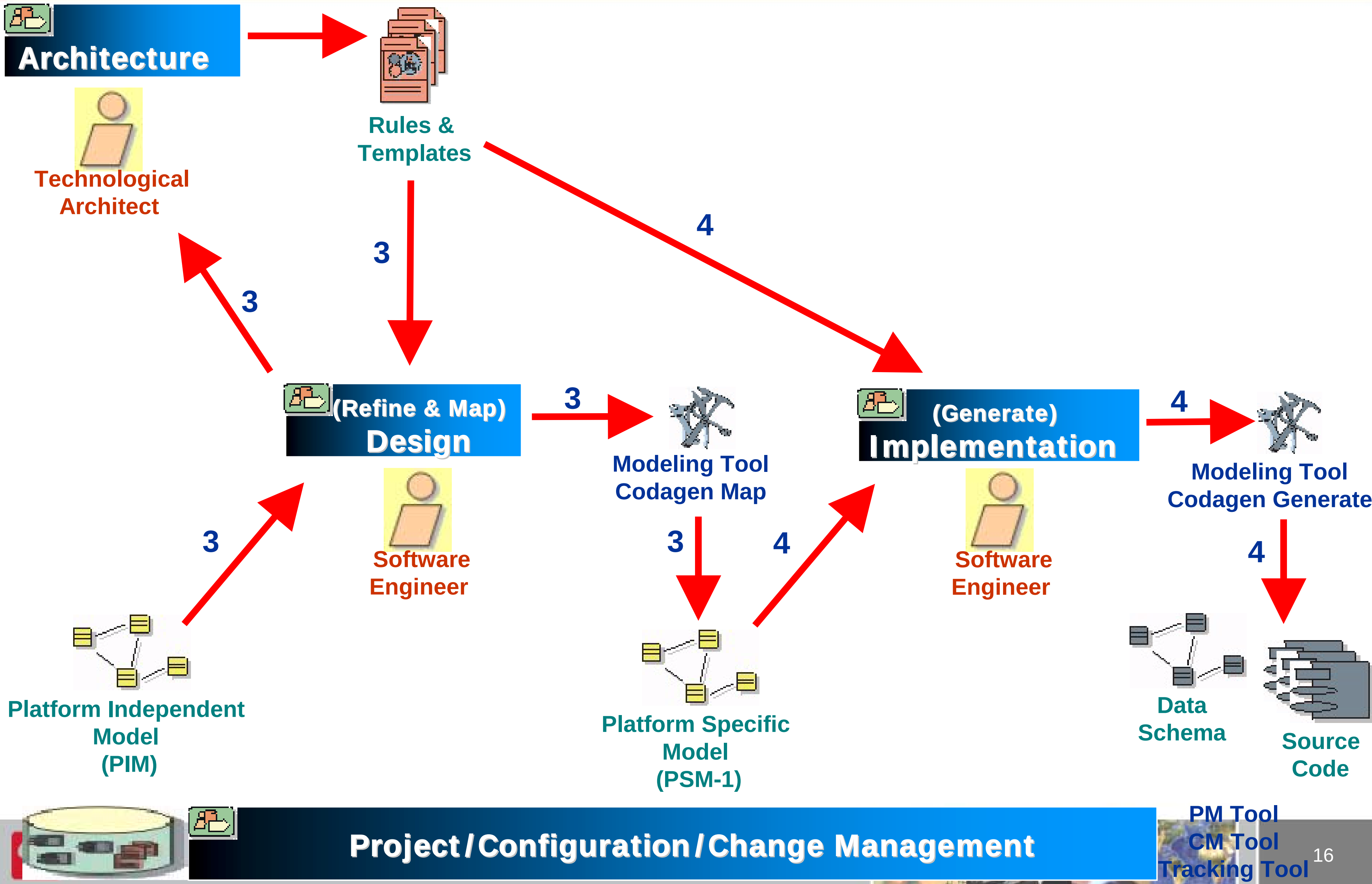

Platform Independant Model (PIM)
Transformation Markers
SPECIFY
UML Profile
MAP
IMPLEMENT
Validation & Transformation Templates
GENERATE
Platform Specific Model (PSM)
Source Code
XMI Model
Documentation
…

Implementation Of The Approach

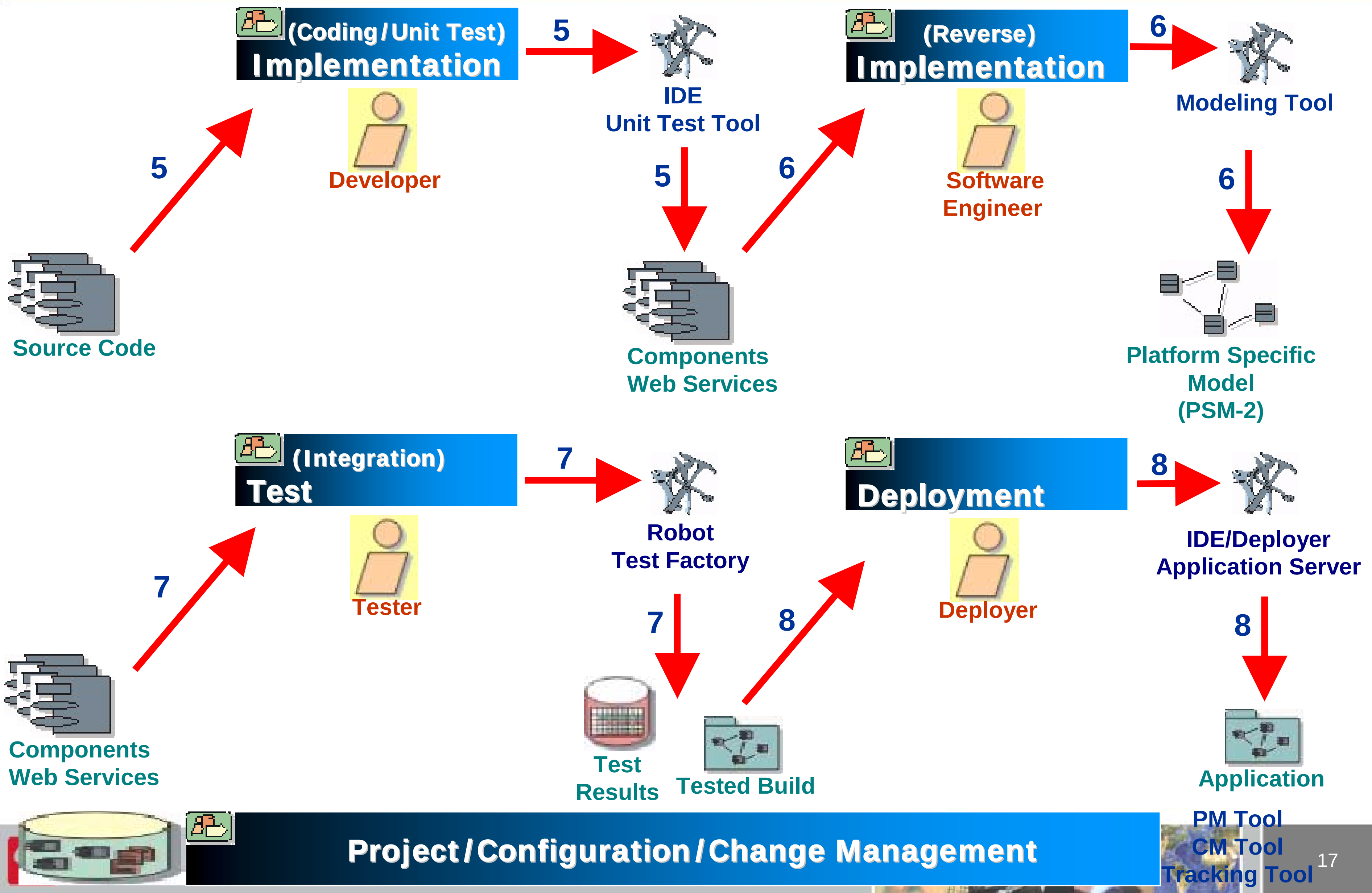
MDA & Inception/Elaboration Phases



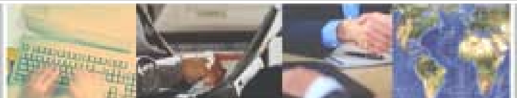
MDA & Elaboration/Construction Phases


Architecture
Rules & Templates
Technological Architect
3
3
4
(Refine & Map) Design
3
Modeling Tool Codagen Map
(Generate) Implementation
4
Modeling Tool Codagen Generate
Software Engineer
3
3
4
Software Engineer
4
Platform Independent Model (PIM)
Platform Specific Model (PSM-1)
Data Schema
Source Code
Project/Configuration/Change Management
PM Tool
CM Tool
Tracking Tool

MDA & Construction/Transition Phases

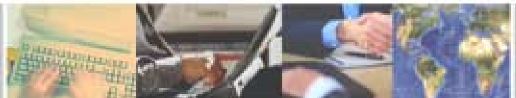


Case Study



Context

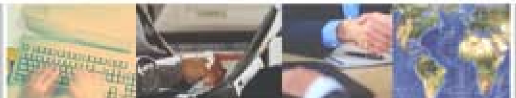
- **Project Description**
  - Re-Create the systems top to bottom with new technologies.
  - Technologies: Java/J2EE & C++/CORBA.
  - Over 52 000 man/day project.
  - Multiple projects in parallel.
  - Very high specifications in term of security, integrity, availability and performance.



Context

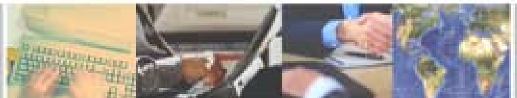
Project's Objectives

Productivity:	Development Cycle
Reusabiltiy:	Architecture, Frameworks, Design Patterns, Components, etc.
Integration:	Approach and Tools Kit
Collaboration:	Between all the interveners of the development cycle
Multi-Projects:	Approach and Tools Kit that support multiple projects realization in parallel.



Context

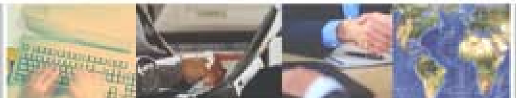
- **Based on industry standards**
  - OMG’s Model-Driven Architecture approach.
  - Rational Unified Process for the development process methodology.
  - UML formalism.
  - Enterprise tools recognised by the industry.



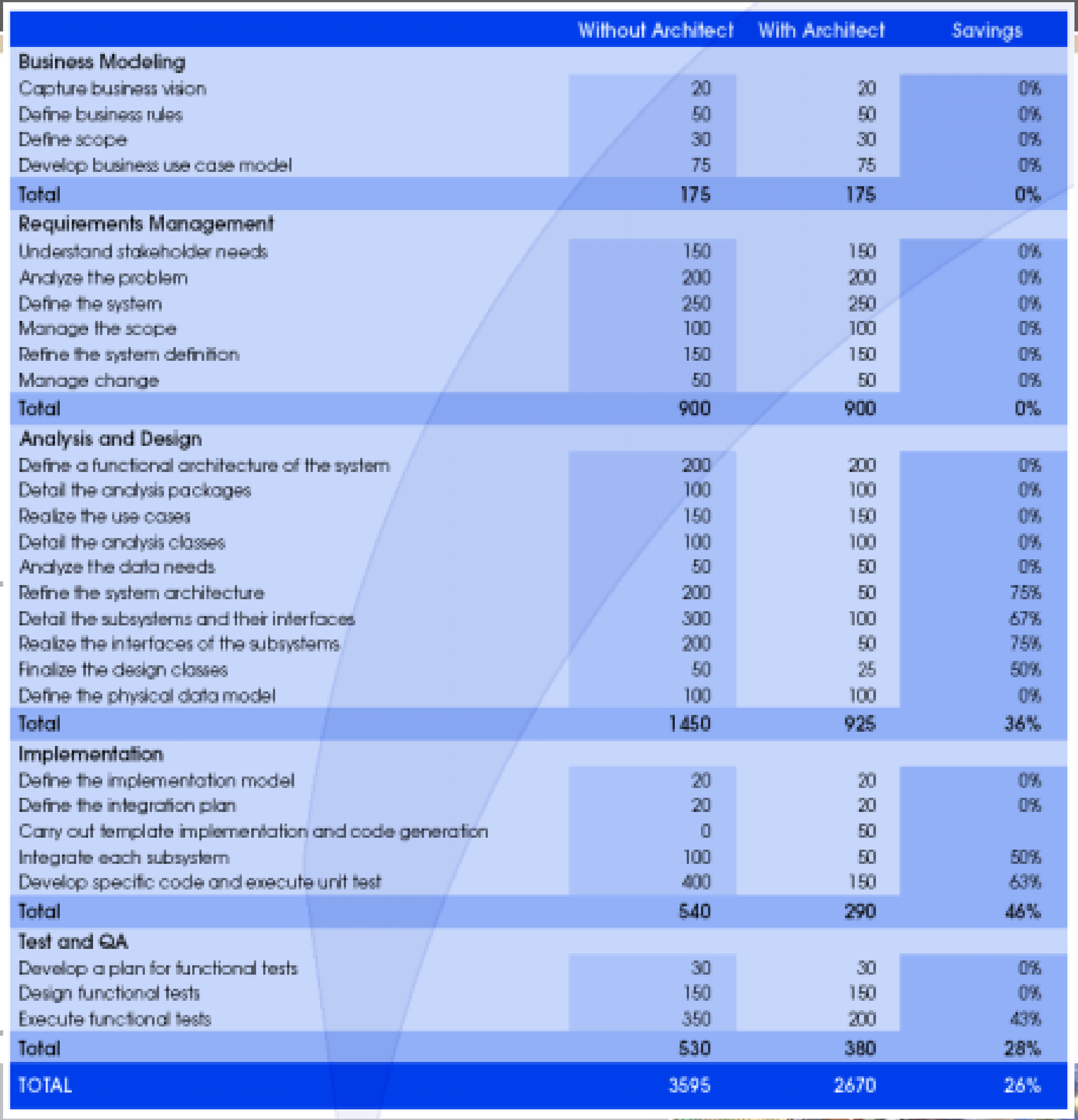
Context

- **Gains of Productivity**
  - Based on reuse and code generation
  - Define benchmarking process and evaluation criterias
  - Code generators evaluation & benchmarking
  - ROI calculation
  - Code generator selection
  - Measurement of productivity gains

  - Selected tools: Codagen



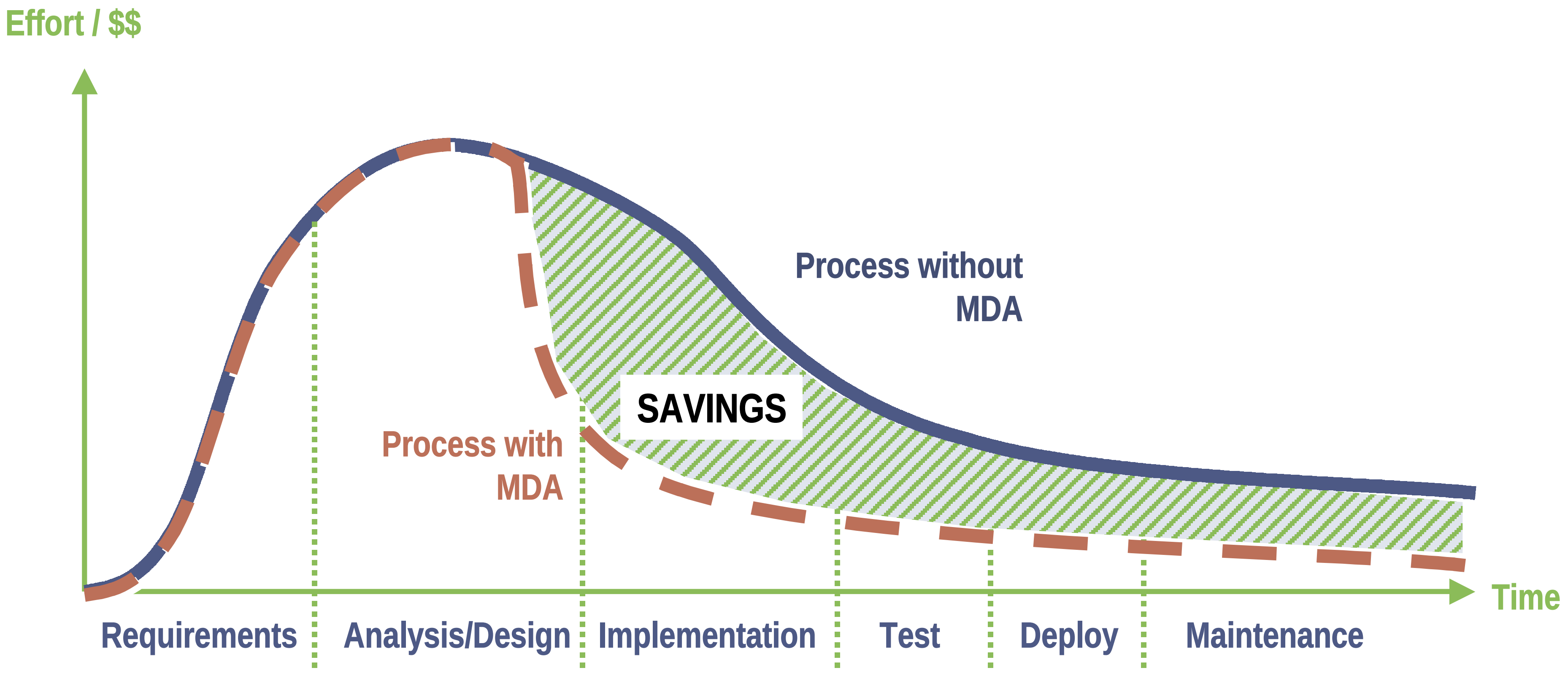
Savings

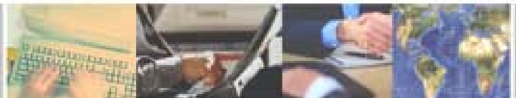


	Without Architect	With Architect	Savings
Business Modeling			
Capture business vision	20	20	0%
Define business rules	50	50	0%
Define scope	30	30	0%
Develop business use case model	75	75	0%
Total	**175**	**175**	**0%**
Requirements Management			
Understand stakeholder needs	150	150	0%
Analyze the problem	200	200	0%
Define the system	250	250	0%
Manage the scope	100	100	0%
Refine the system definition	150	150	0%
Manage change	50	50	0%
Total	**900**	**900**	**0%**
Analysis and Design			
Define a functional architecture of the system	200	200	0%
Detail the analysis packages	100	100	0%
Realize the use cases	150	150	0%
Detail the analysis classes	100	100	0%
Analyze the data needs	50	50	0%
Refine the system architecture	200	50	75%
Detail the subsystems and their interfaces	300	100	67%
Realize the interfaces of the subsystems	200	50	75%
Finalize the design classes	50	25	50%
Define the physical data model	100	100	0%
Total	**1450**	**925**	**36%**
Implementation			
Define the implementation model	20	20	0%
Define the integration plan	20	20	0%
Carry out template implementation and code generation	0	50	
Integrate each subsystem	100	50	50%
Develop specific code and execute unit test	400	150	63%
Total	**540**	**290**	**46%**
Test and QA			
Develop a plan for functional tests	30	30	0%
Design functional tests	150	150	0%
Execute functional tests	350	200	43%
Total	**530**	**380**	**28%**
TOTAL	**3595**	**2670**	**26%**

Major Savings

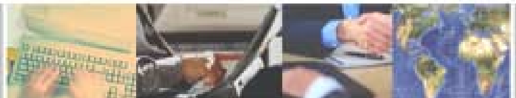
Savings


Effort / $$
Process without MDA
SAVINGS
Process with MDA
Time
Requirements
Analysis/Design
Implementation
Test
Deploy
Maintenance



Benefits

- Concrete gains of productivity
  - Decrease time to market
  - Cost savings
- Major reduction of the development team
  - Few seniors (architects) define, implement and maintain the architecture
  - UML Modelers define and maintain the business
  - Few junior developers code the specific (business rules and validation)
  - Don’t need anymore an army of senior or specialist developers
- Homogeneous & uniform architecture
- Simple development process
  - Very well defined activities and roles
  - Very well defined artifacts and deliverables to produce



Questions & Discussion