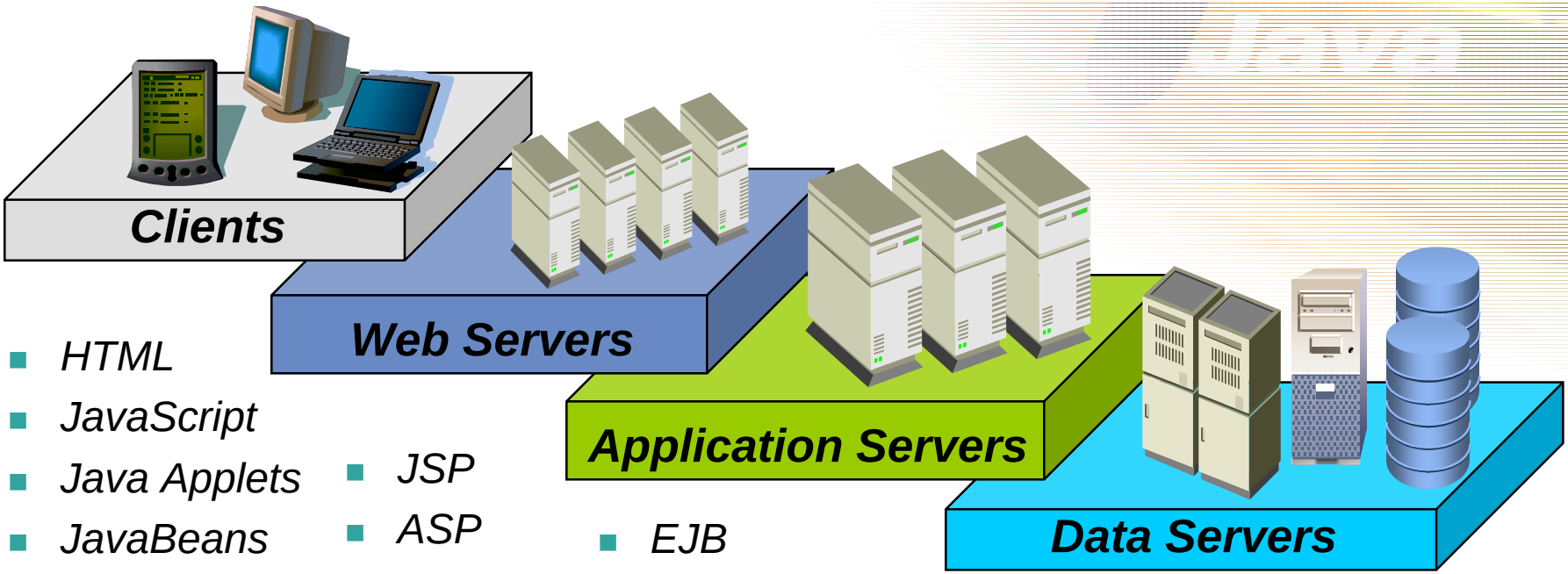
A Practical Approach to MDA

Paul W. Styles
Chief Architect – Strategic Planning

COMPUWARE.

Multi-tier Application Architecture



- *HTML*
- *JavaScript*
- *Java Applets*
- *JavaBeans*

- *JSP*
- *ASP*
- *XML*
- *WML*

- *EJB*
- *Java Servlets*
- *CORBA*
- *DCOM*
- *ODBC/JDBC*

- *SQL*
- *Stored Procedures*

COMPUWARE.

Java/J2EE Application Development Challenges

- Rapidly respond to business change – Time to market
- Increase developer productivity
- Maximize application quality and reliability – Minimize risk
- Maintain currency with new versions of technology
- Leverage existing investments
- Enforce the use of best practices, standards and guidelines

COMPUWARE.

Enterprise Application Development

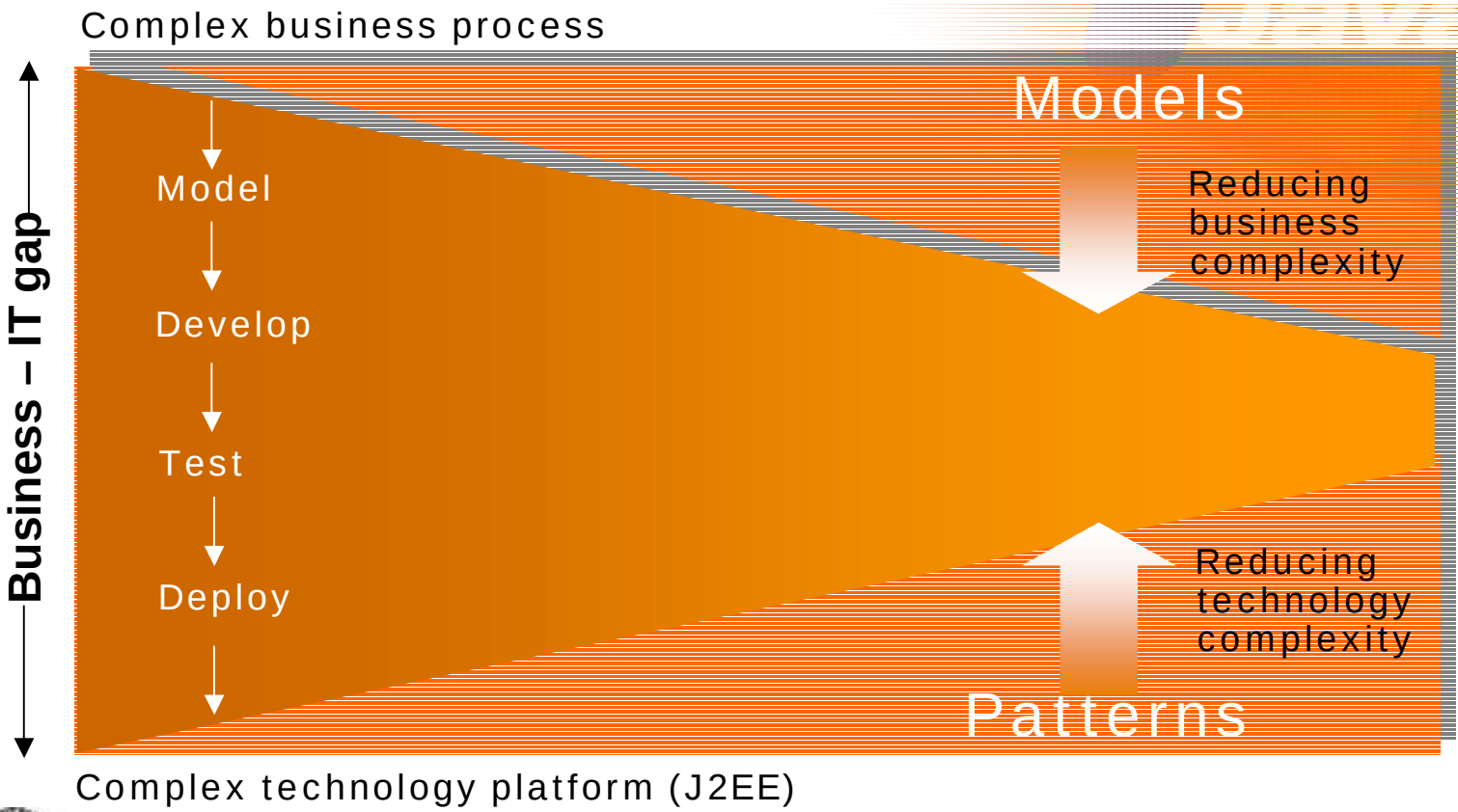
The reality…

"We try to solve the problem by rushing through the design process so that enough time is left at the end of the project to uncover the errors that were made because we rushed through the design process"

Glenford J. Myers

COMPUWARE.

Closing the Gap
The need for Models and Patterns



COMPUWARE.

A New Development Paradigm
Model-driven pattern-based

- Companies that want to maintain or increase their future competitive edge will need to begin evaluating, planning for and migrating development staff to at least one of the two alternative and more efficient forms of development, model-driven pattern-based (MDPB) or component assembly and orchestration (CAO).

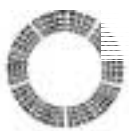
Giga Information Group®

John Meyer
Sr. Industry Analyst

Gartner Inc.

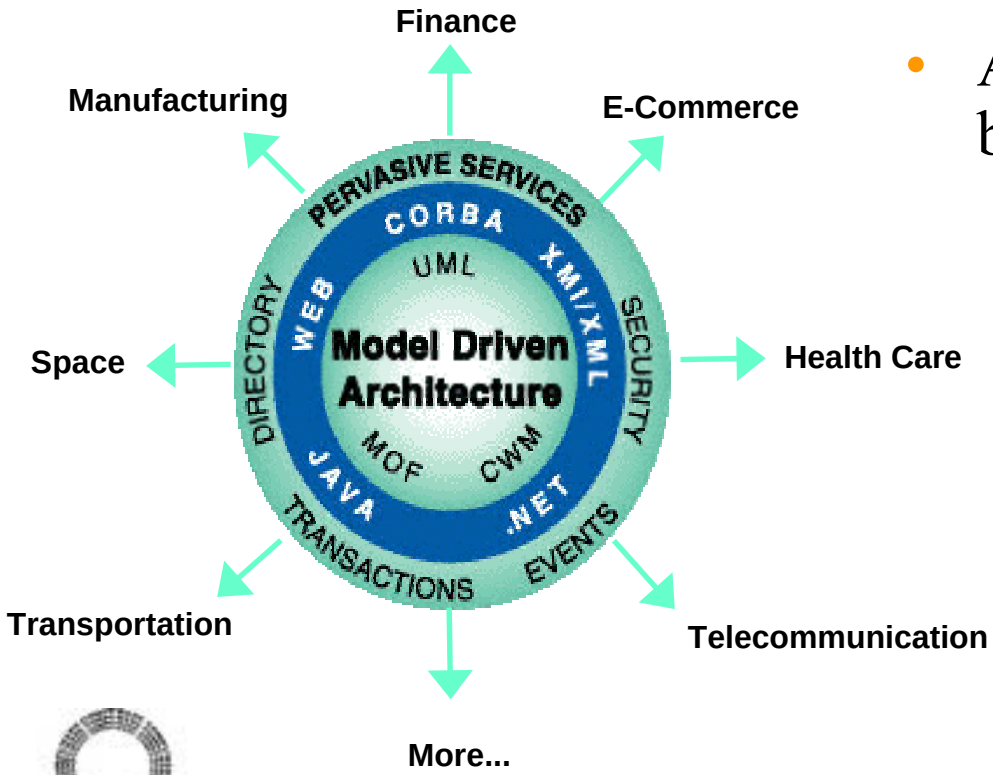
Michael Blechar
VP Internet and e-Business Technologies

- Organizations using a model-driven or pattern-based application development framework containing a large inventory of business components have the potential to be five to ten times more productive and responsive than those that do not



COMPUWARE.

Object Management Group
Model-Driven Architecture (MDA)

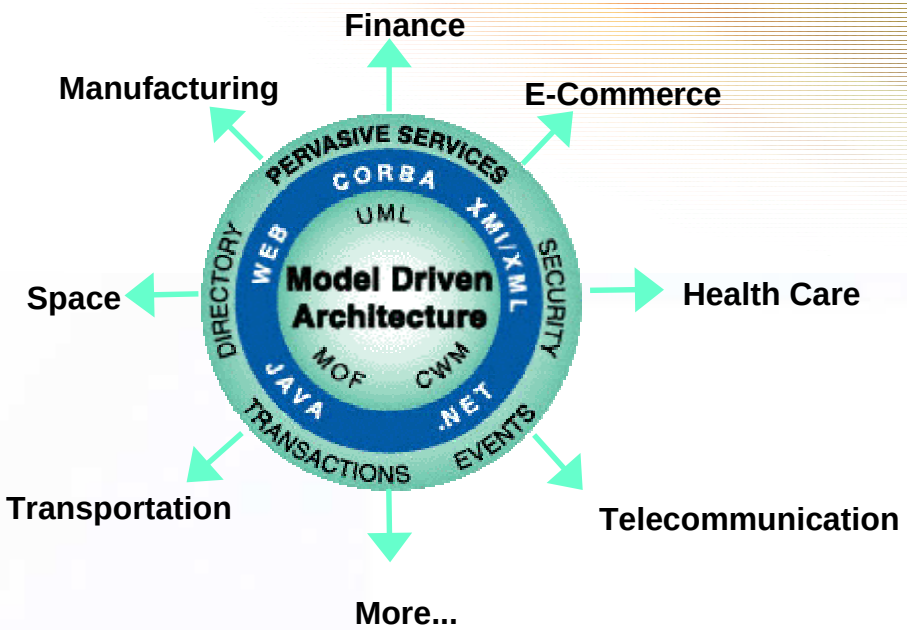


- A new way to specify and build systems
  - Based on modeling with UML
  - Modeling instead of programming
  - Merging modeling and coding
  - Based on standards
    - UML, MOF, CWM, XMI….

COMPUWARE.

MDA Qualities

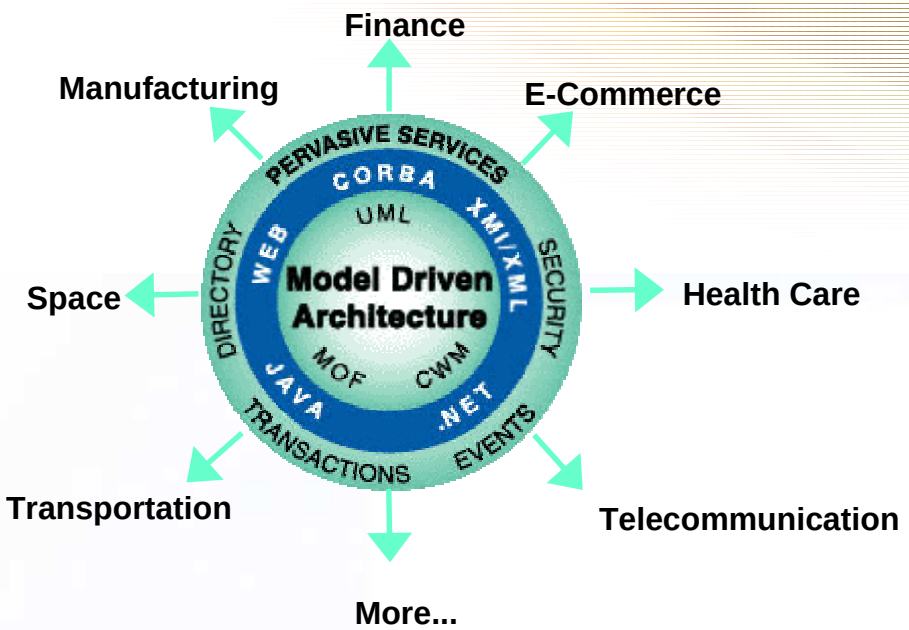
- **Portability**
- **Cross-platform Interoperability**
- **Platform Independence**
- **Domain Specificity**
- **Productivity**



COMPUWARE

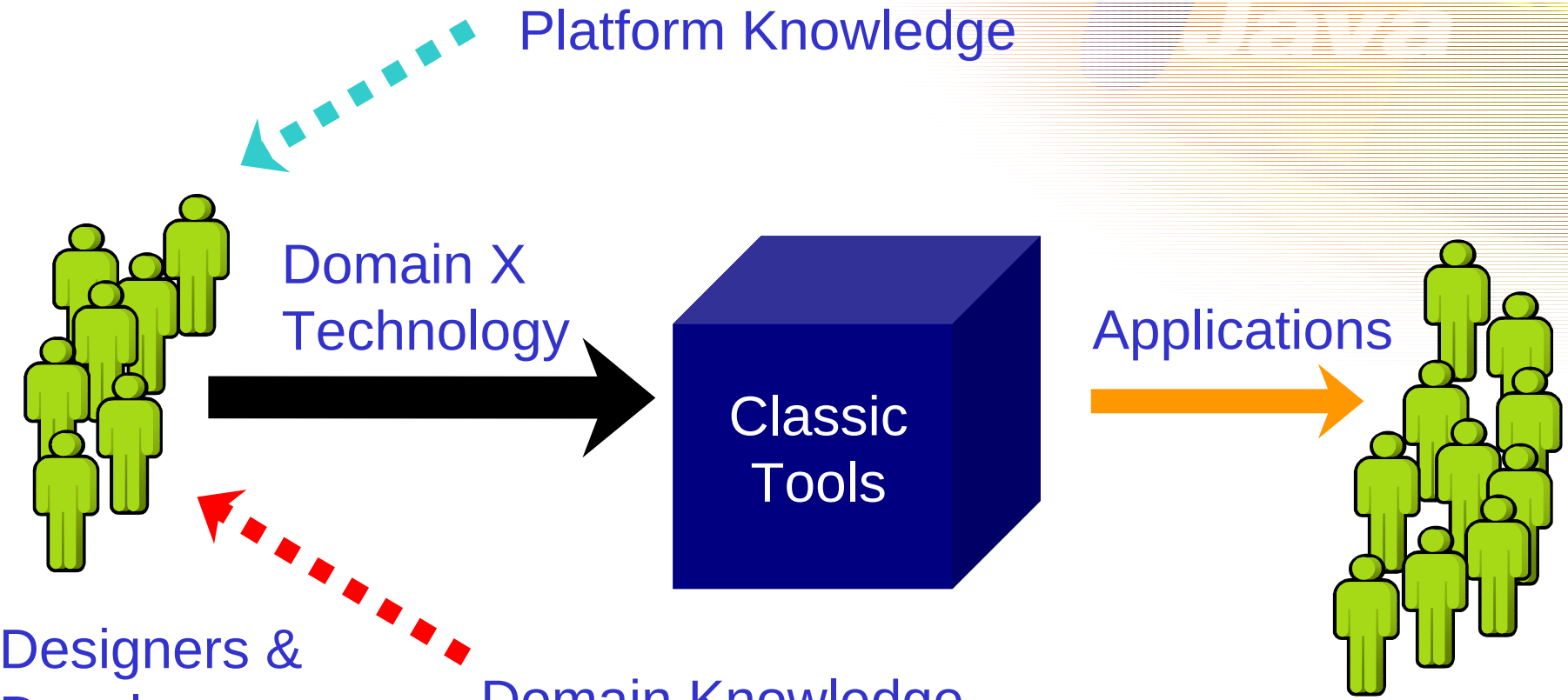
MDA Benefits

- **Reduced cost**
- **Reduced development time**
- **Improved application quality**
- **Increased return on IT investments**
- **Rapid inclusion of emerging technologies**



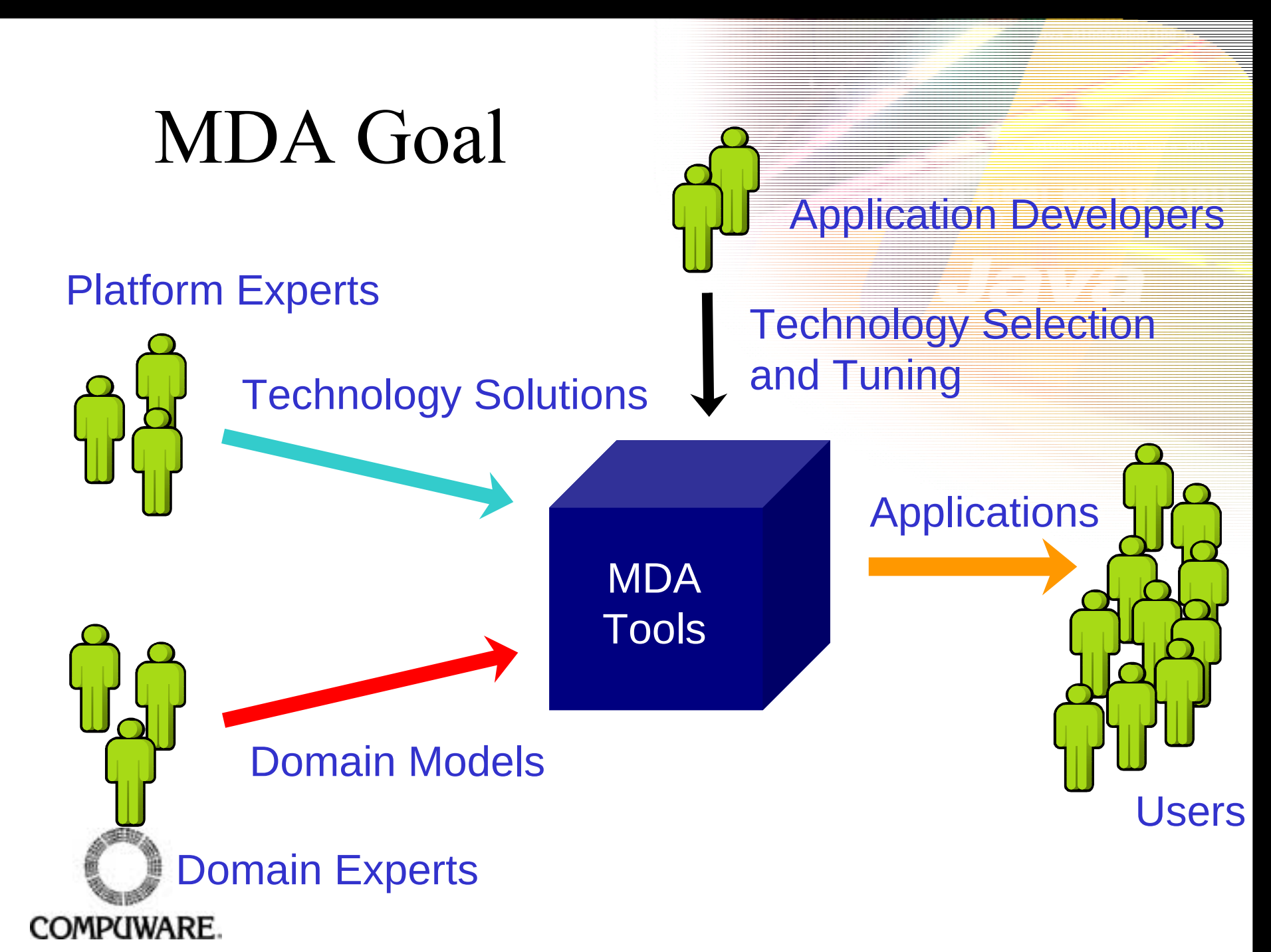
COMPUWARE

Classic Modeling and Development



COMPUWARE.

MDA Goal



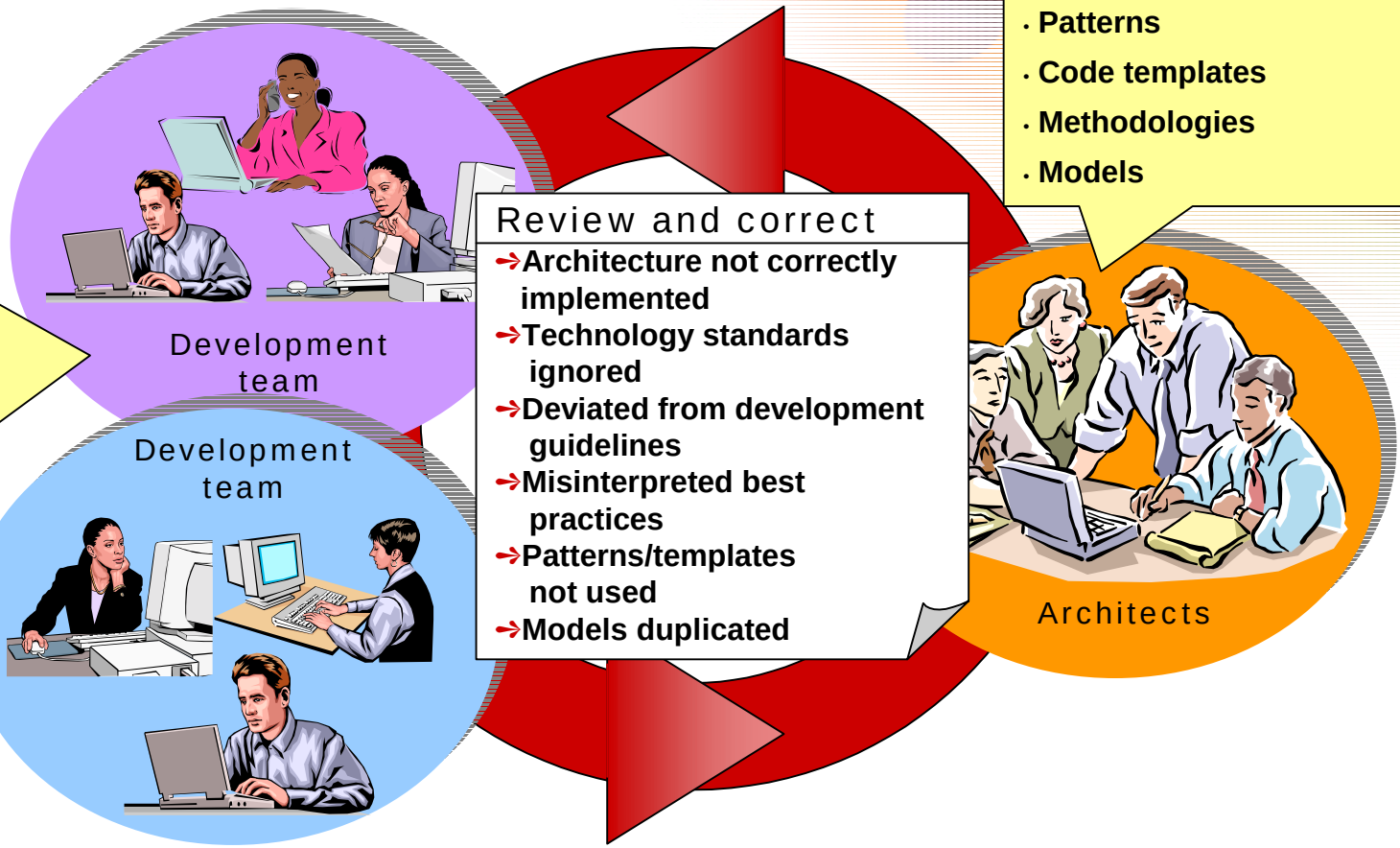
COMPUWARE.

Knowledge transfer

Ensuring the use of best practices, standards & guidelines

Define

- Overall application architecture
- J2EE expertise
- Technology standards
- Development guidelines
- Best practices
- Patterns
- Code templates
- Methodologies
- Models



Review and correct

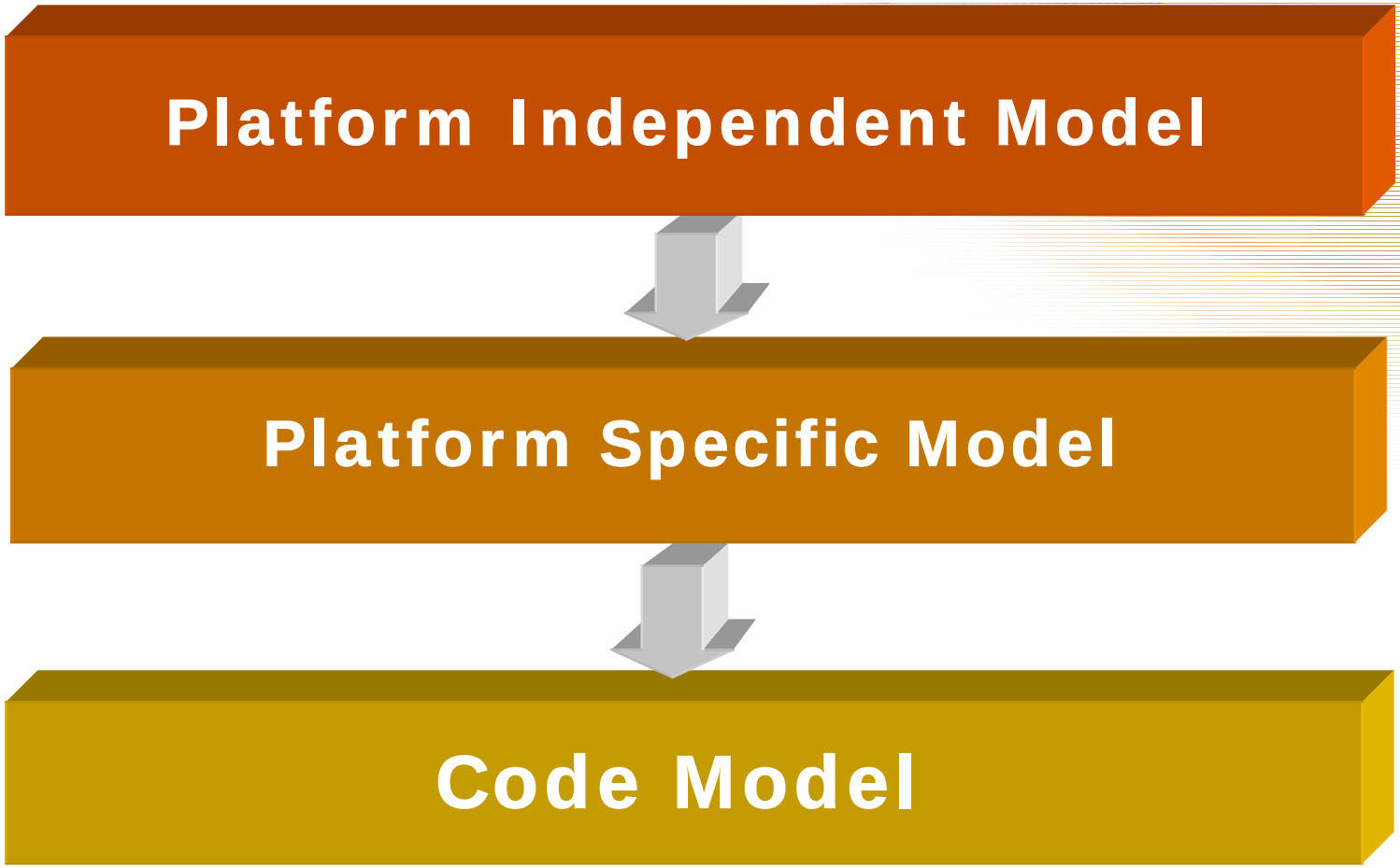
- ➔Architecture not correctly implemented
- ➔Technology standards ignored
- ➔Deviated from development guidelines
- ➔Misinterpreted best practices
- ➔Patterns/templates not used
- ➔Models duplicated

Implement

- Application architecture
- Technology standards
- Development guidelines
- Best practices
- Patterns
- Code templates
- Methodologies
- Models

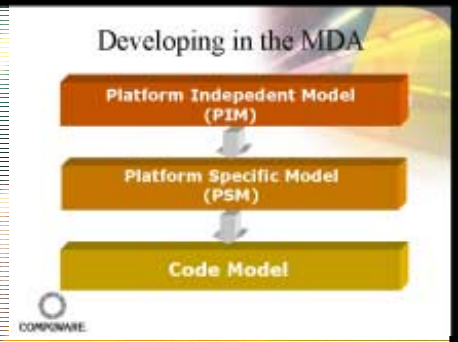
COMPUWARE.

Model Driven Architecture

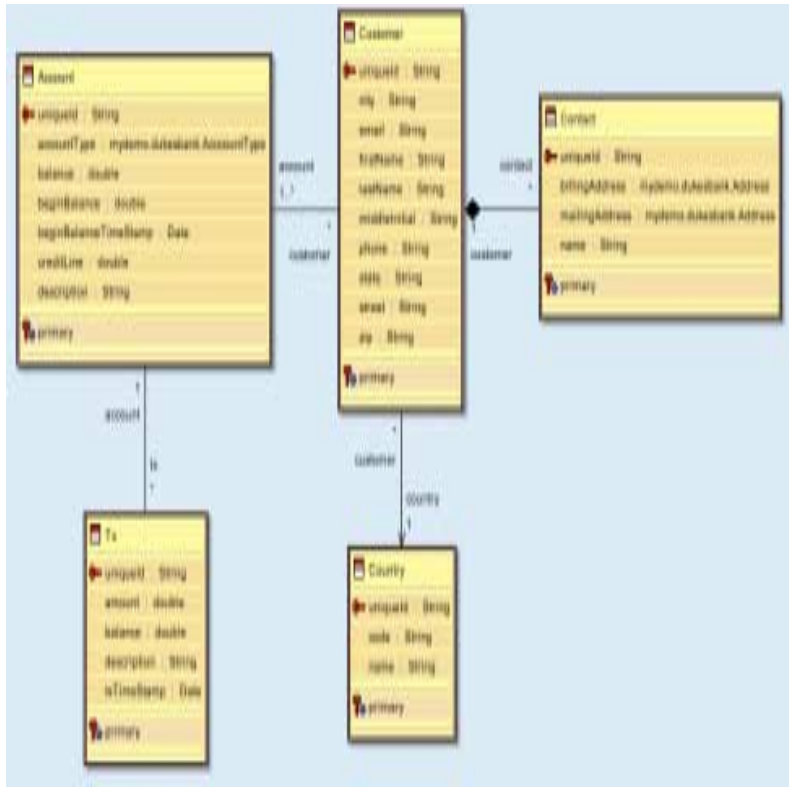


COMPUWARE.

Platform Independent Model
Step one

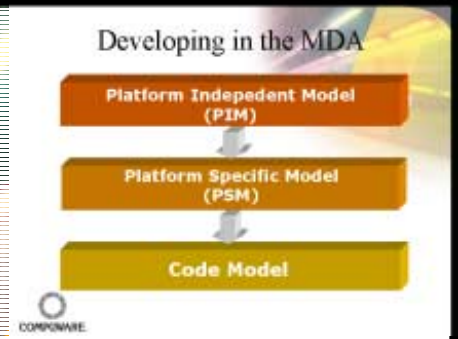


- Expressed in the UML
  - Structure
  - Behavior
- Develop at a high abstraction level
  - Undistorted by technical details
  - Business centric
  - Focus on application functionality
- Enrich with Business rules:
  - Constraints
  - Expressions
- Automated reuse of model definitions and business rules
- Change applications rapidly in the model

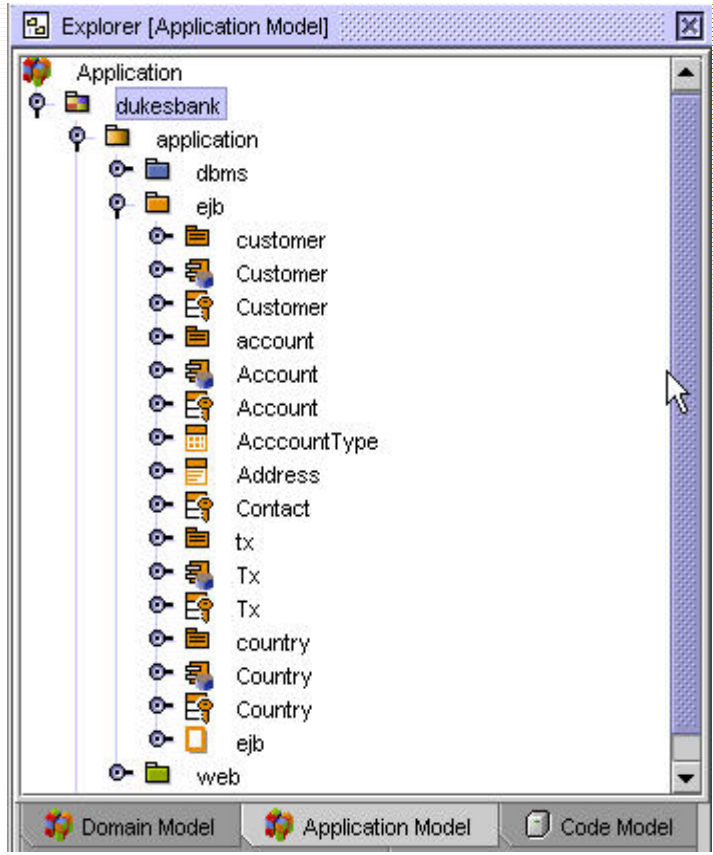


COMPUWARE.

Platform Specific Model
Step two

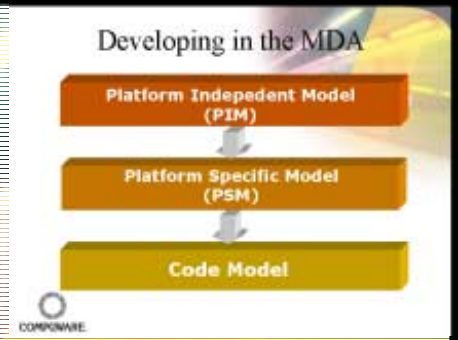


- Automatic transformation from PIM to PSM (Selected target platform)
  - Consistency
  - Quality
- Presentation Model (Web)
  - Data schemas, web components, etc.
- Business Logic Model (EJB)
  - Data schemas, key classes, entity components, session components, etc.
- Data Model (DBMS)
  - Relational data schema
  - Tables, columns, keys, etc.

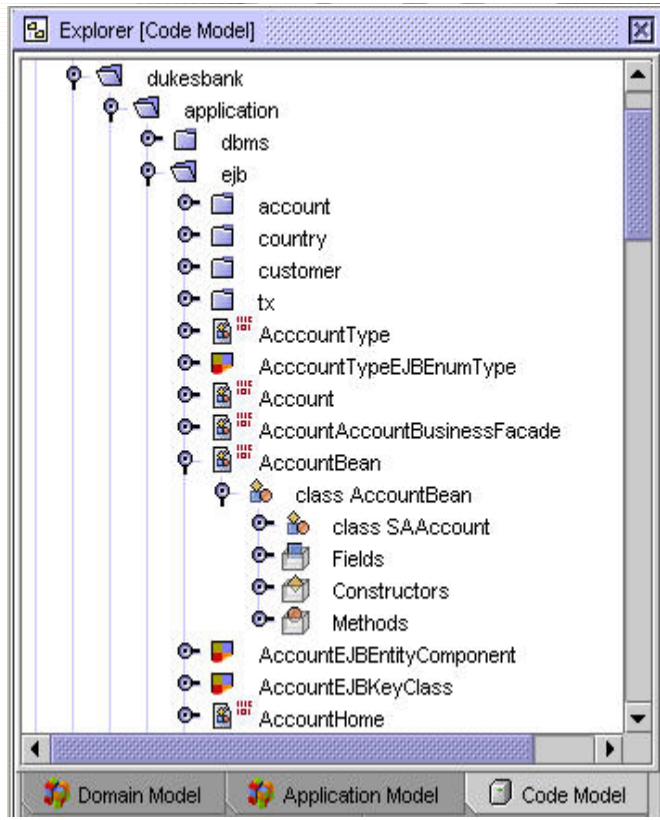


COMPUWARE.

Code Model
Step three



- Automatic transformation from PSM to Code
  - Productivity
  - Quality
  - Consistency
- Presentation Tier
  - Model View Controller framework (MVC)
  - Java Server Pages (JSP), servlets
- EJB Tier
  - Remote, bean class, home and primary key classes etc.
  - According to EJB specification
- DBMS Tier
  - SQL scripts
- Application deployment descriptors
  - For target application server



COMPUWARE.

Implementing a Model Driven Architecture

- How to transform model within the MDA and realize the benefits of:
  - Productivity
  - Portability
  - Interoperability

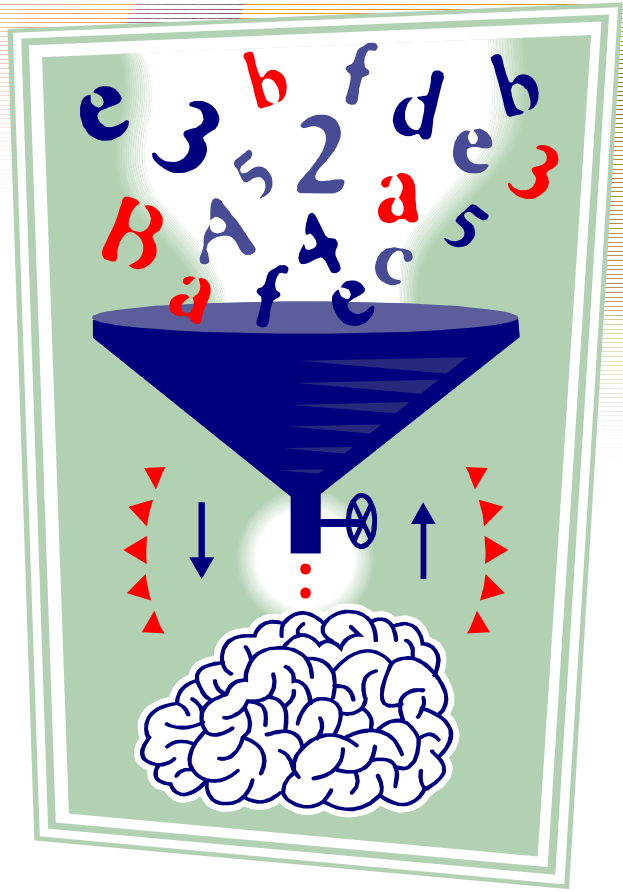
Pattern Automation!

Why Patterns?

- Software Architects usually have clear ideas on how to implement a given model, based on:
  - Internal standards, guidelines and best practices
  - Coding and architectural standards
  - Past experience
  - Existing libraries of code

Benefits of Patterns

- Make best use of experienced developers
- Reduces learning curve
- Reusable, application-independent components
- Shields developers from most of the low-level coding details
- Patterns generate consistent and high quality code across multiple developers and applications
- Generated code is pre-tested and is therefore more reliable
- Standards and enforcement of standards



COMPUWARE.

Technology Change

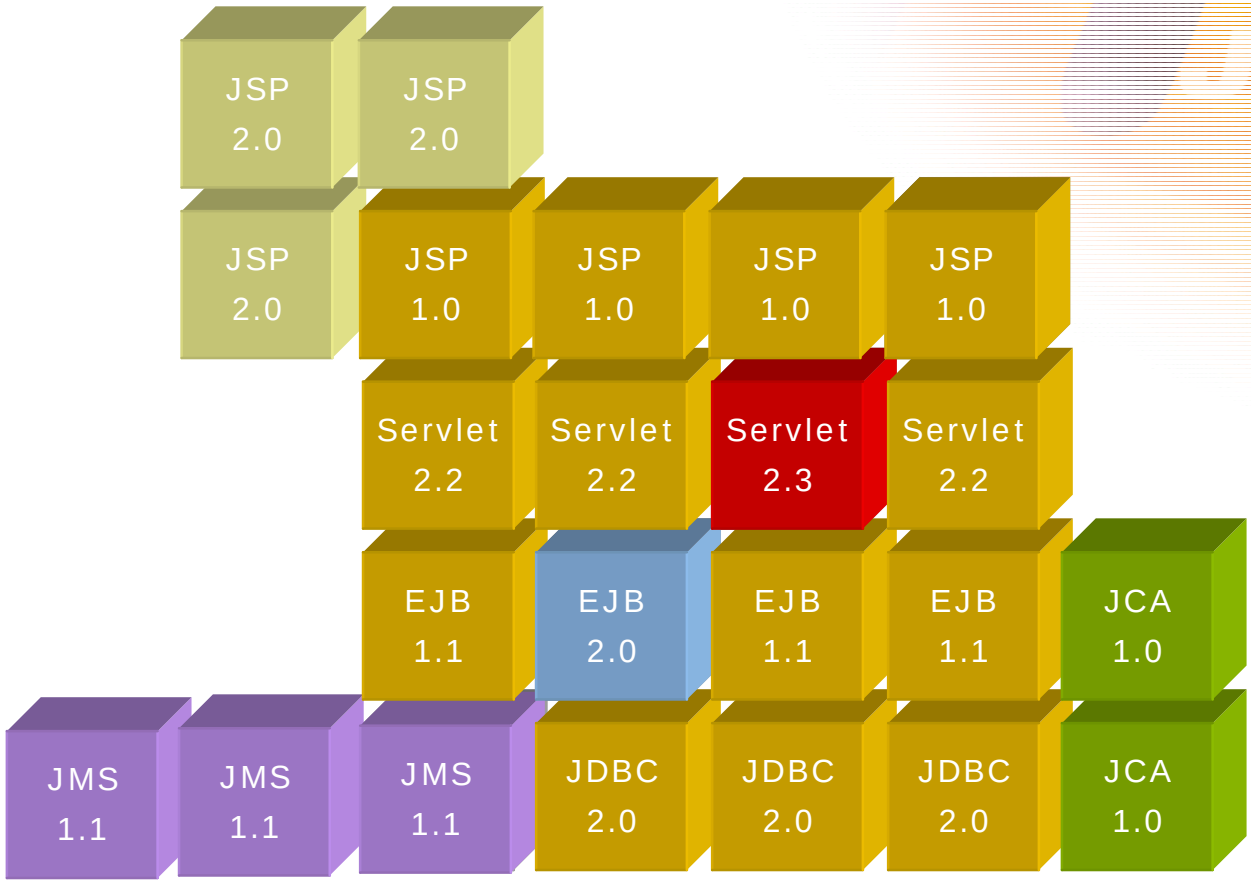
Keeping up with the pace of J2EE

Version 1 of the application

COMPUWARE.

Technology Change

Keeping up with the pace of J2EE

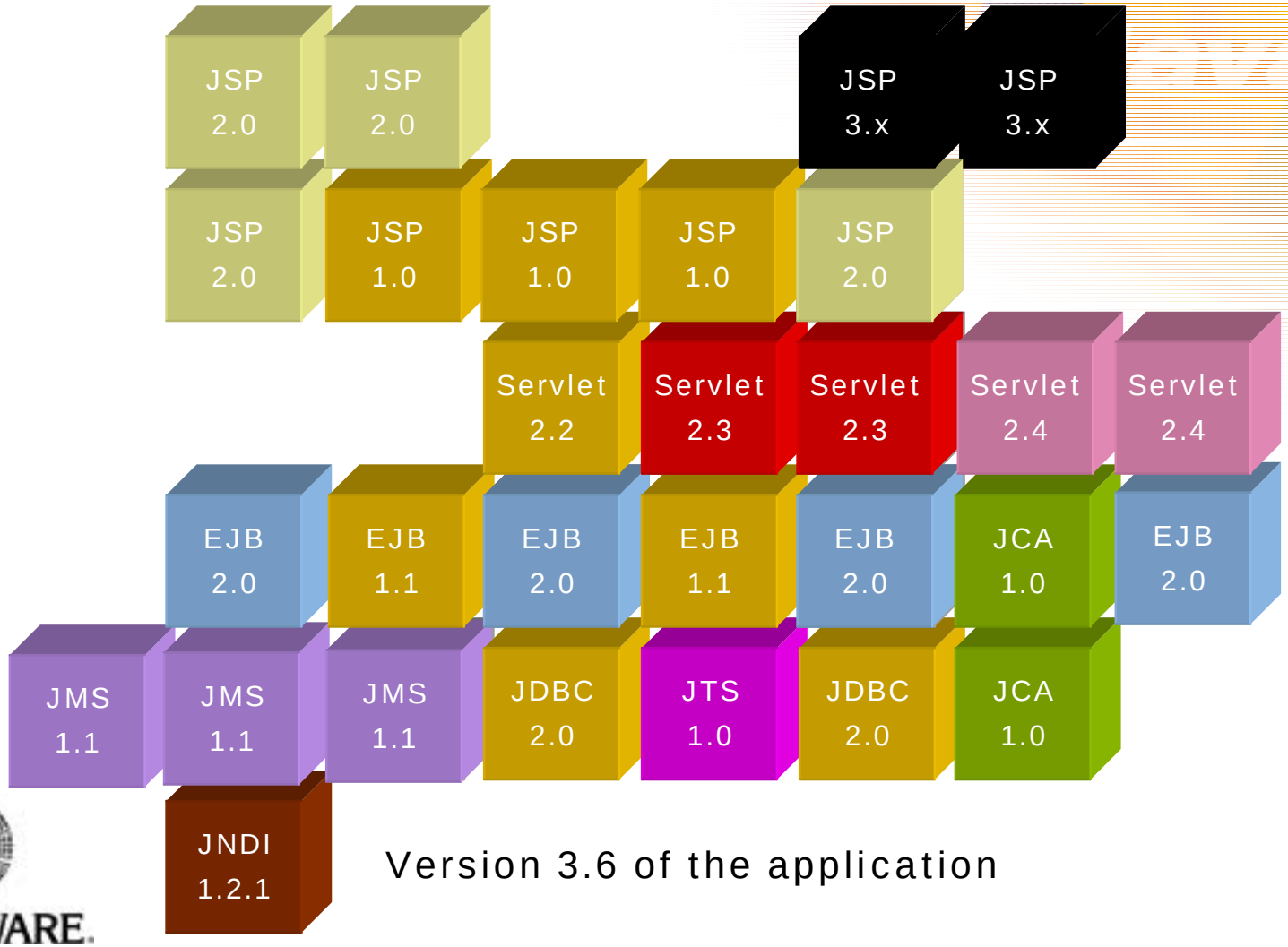


Version 2.3 of the application

COMPUWARE.

Technology Change

Keeping up with the pace of J2EE

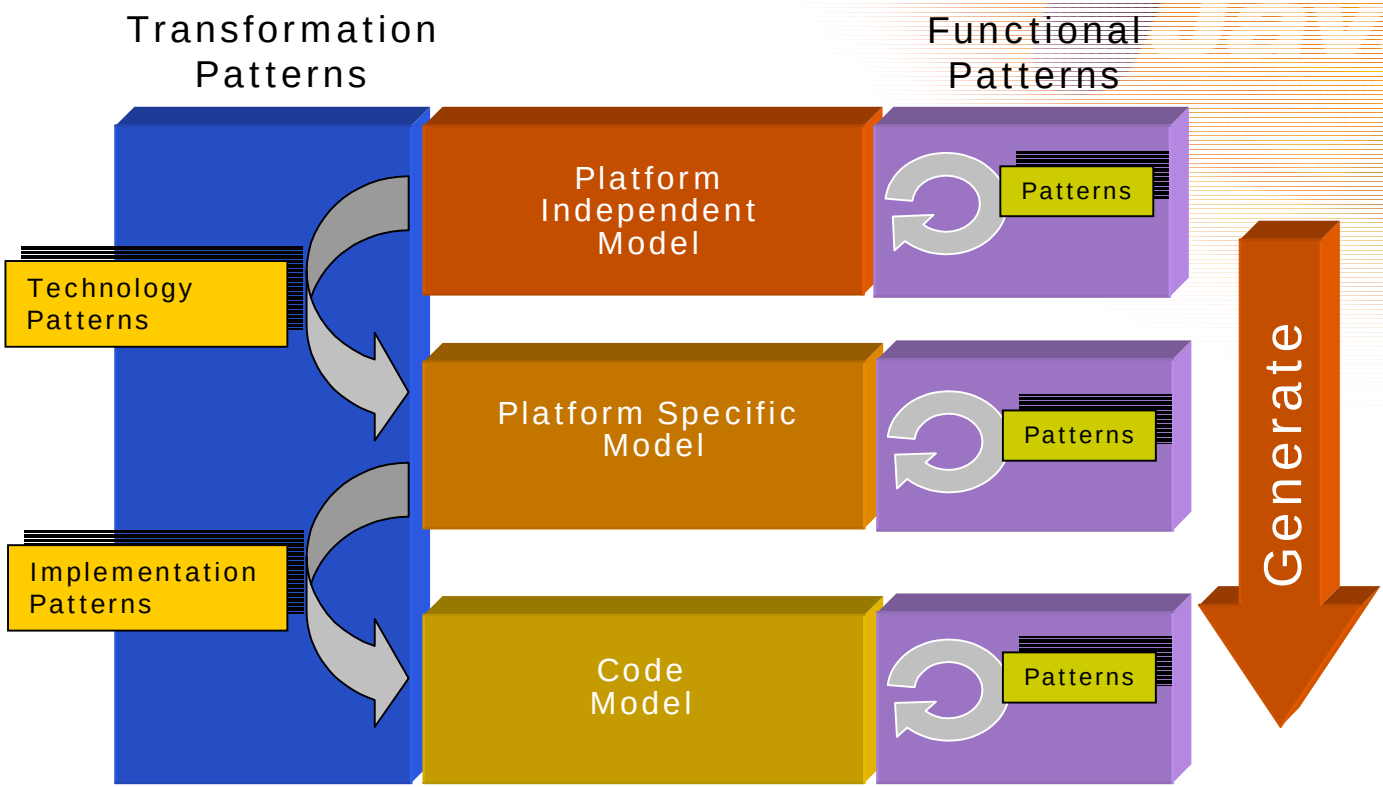


Version 3.6 of the application

COMPUWARE.

Pattern Driven Generation

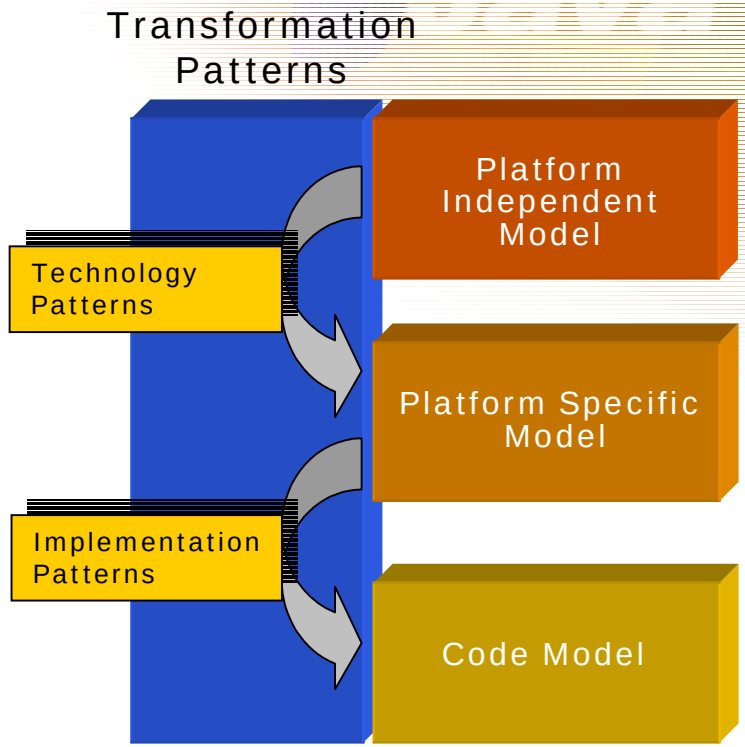
Model & Patterns



COMPUWARE.

Customizing and maintaining patterns

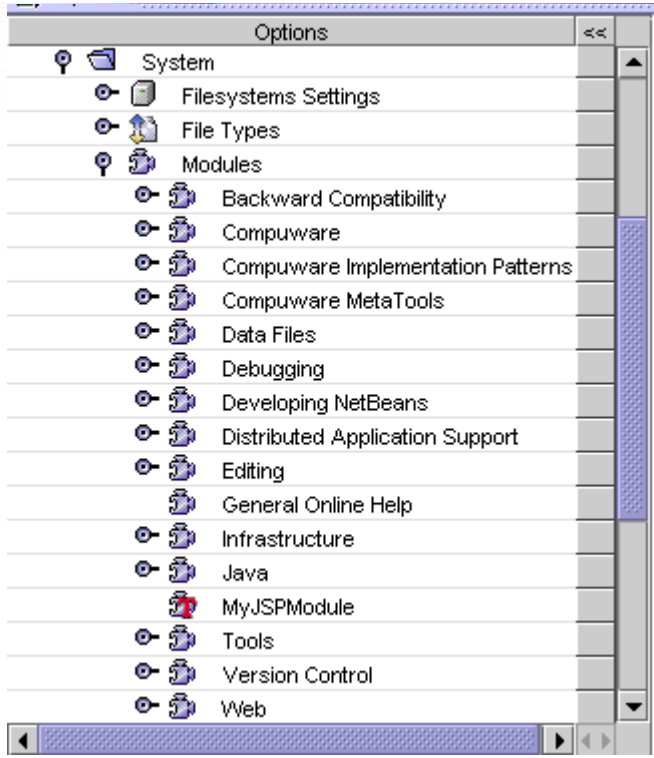
- Allow organizations to implement their internal coding standards
- Automatically transfers expert knowledge to designers and developers
- Ensures designers/developers implement the standards
- Full control over generated code
- Experienced J2EE architects can add, update and delete the Patterns



COMPUWARE.

Pattern development process

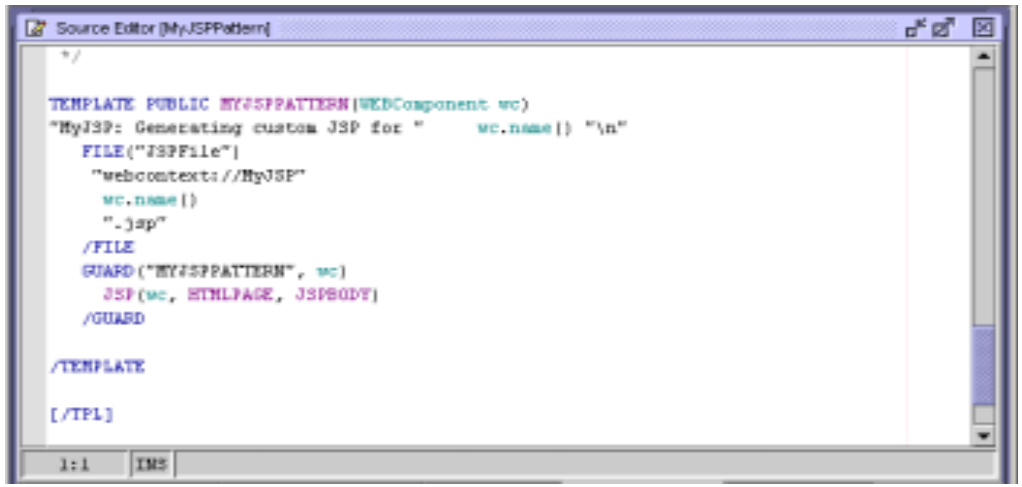
1. Determine requirements for the pattern
2. Determine which of the MDA models provide a suitable basis for the pattern
3. Relate requirements to the model
4. Implement the pattern using a pattern language
5. Deploy the pattern as a module



COMPUWARE.

Pattern extension tools

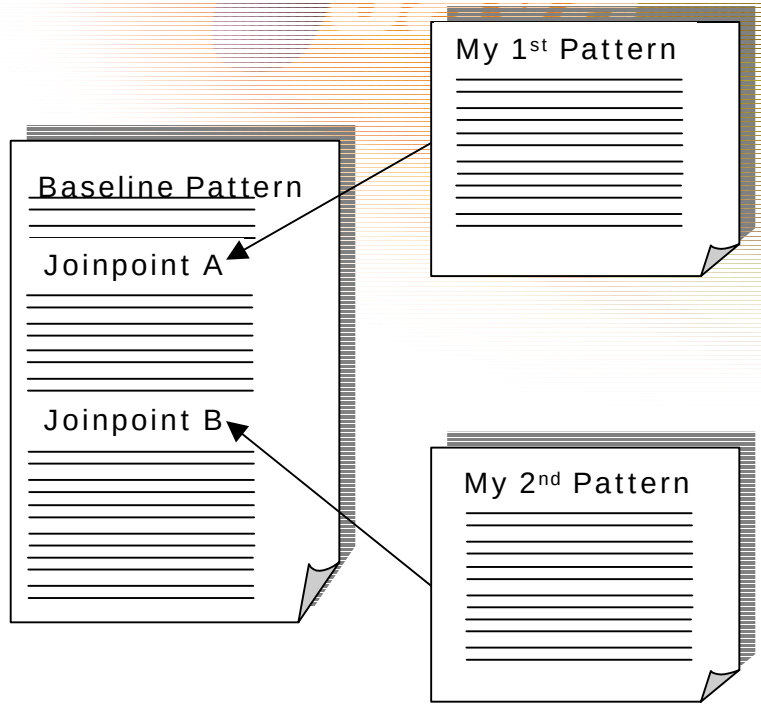
- Pattern authoring workbench
- Pluggable pattern architecture
- Platform independent pattern language
  - Pattern language to implement logic and apply the correct patterns
  - Access to the Meta Object Facility (MOF) repository
  - Small language compiled into target platform code



COMPUWARE.

Extension of patterns

- Provide pattern sources
  - ✔ Customer has all flexibility
  - ✘ Customer doesn't benefit from upgrades
- Provide hooks (Joinpoints) that allow customers to override and extend the default pattern functionality
  - ✔ Customer has sufficient flexibility
  - ✔ Customer benefits from upgrades



COMPUWARE.

Are you *writing* code or *building* an application?

"Building an application"

- Focus on functionality and behavior ("what"), not on implementation details ("how")
- Pattern based generation
  - Quality derived from repeatability
  - Industrial approach to software development
  - Enforcing of standards and best practices
- Easy maintenance: model-centric synchronization and pattern based re-generation

Focus on Business instead of Technology

MDA Essentials

- Embracement and appreciation of existing IT technologies
- Separation between, and reusability of, domain and platform expertise
- Quick adaptability of domain and technology changes
- Generation of working high-quality applications and integrations

COMPUWARE.