



Model Driven Development
Through the Agile Looking Glass



Agile India, 5-6 May, 2006, Bangalore, India
egottesman@sapient.com

Objectives

- Present Sapien's point-of-view on emerging software development trends
- Discuss how Agile and MDD are together changing the way we develop



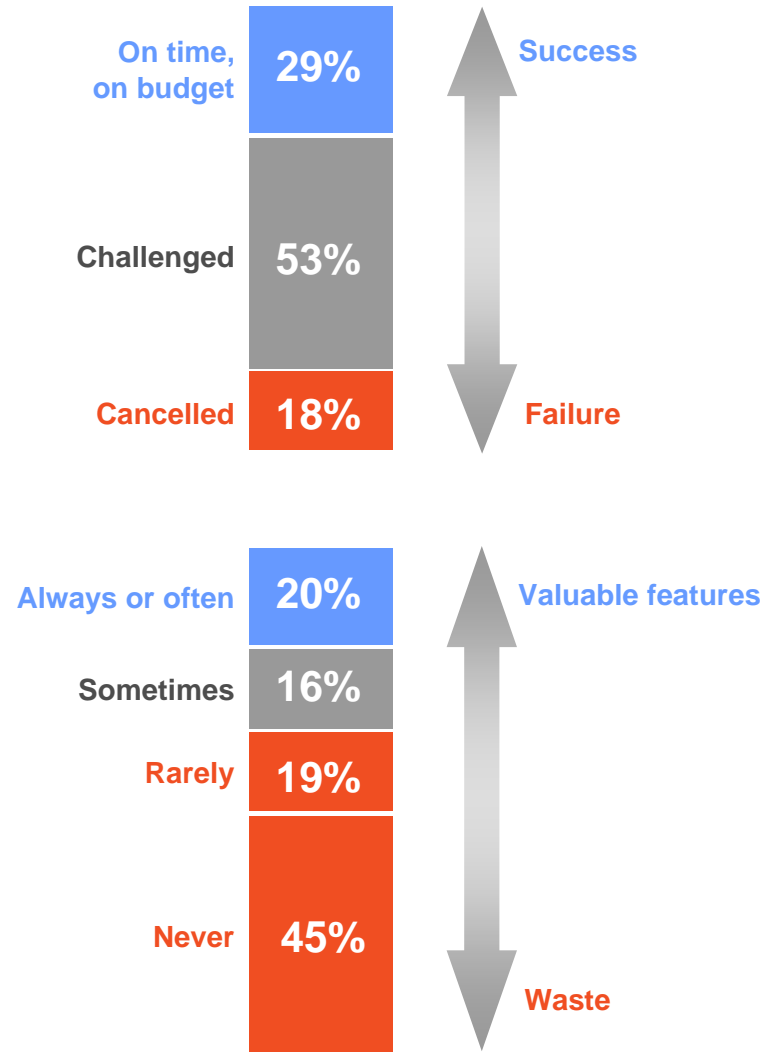
Agenda

- A look at the current state of the industry
- What is the problem we are trying to solve?
- Our solution
- Pilot projects, results and benefits
- Questions and feedback



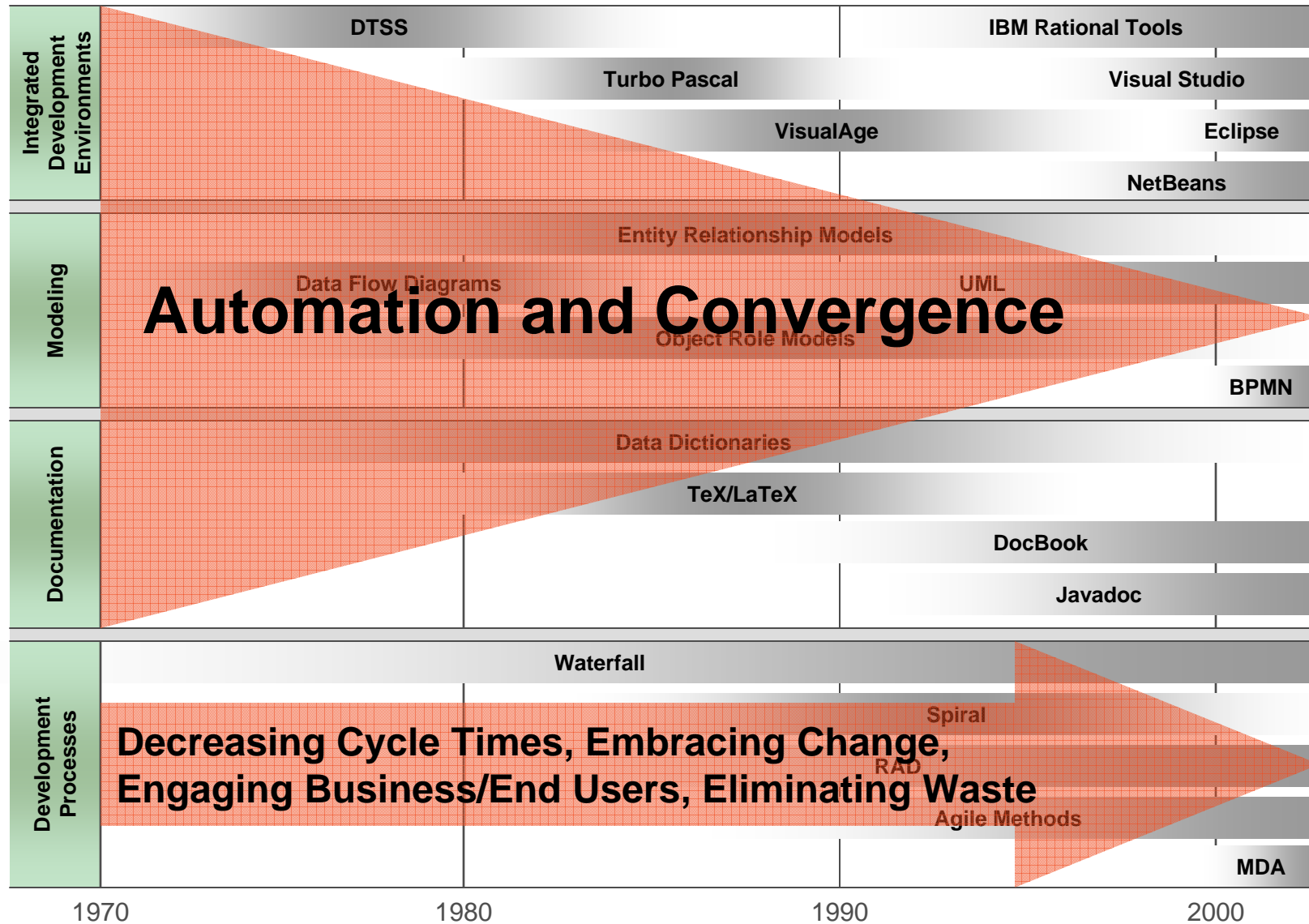
How is our industry performing?

- The vast majority of projects still do not complete on time or on budget
- For solutions that are actually delivered, nearly **two-thirds of the features built are rarely or never used** – in other words, waste!
- The root cause of this waste is fundamentally driven by **the ways that partners engage** with their clients
- Customers have been burned and **aren't willing to wait** before seeing ROI – they want to know the solution will meet their needs



Source: *The Chaos Chronicles*, The Standish Group, 2004.

A Brief (Abridged) History of Software Development...



A common way to develop custom applications today...



Gather requirements



Create models – business process models, object models, data models, ...



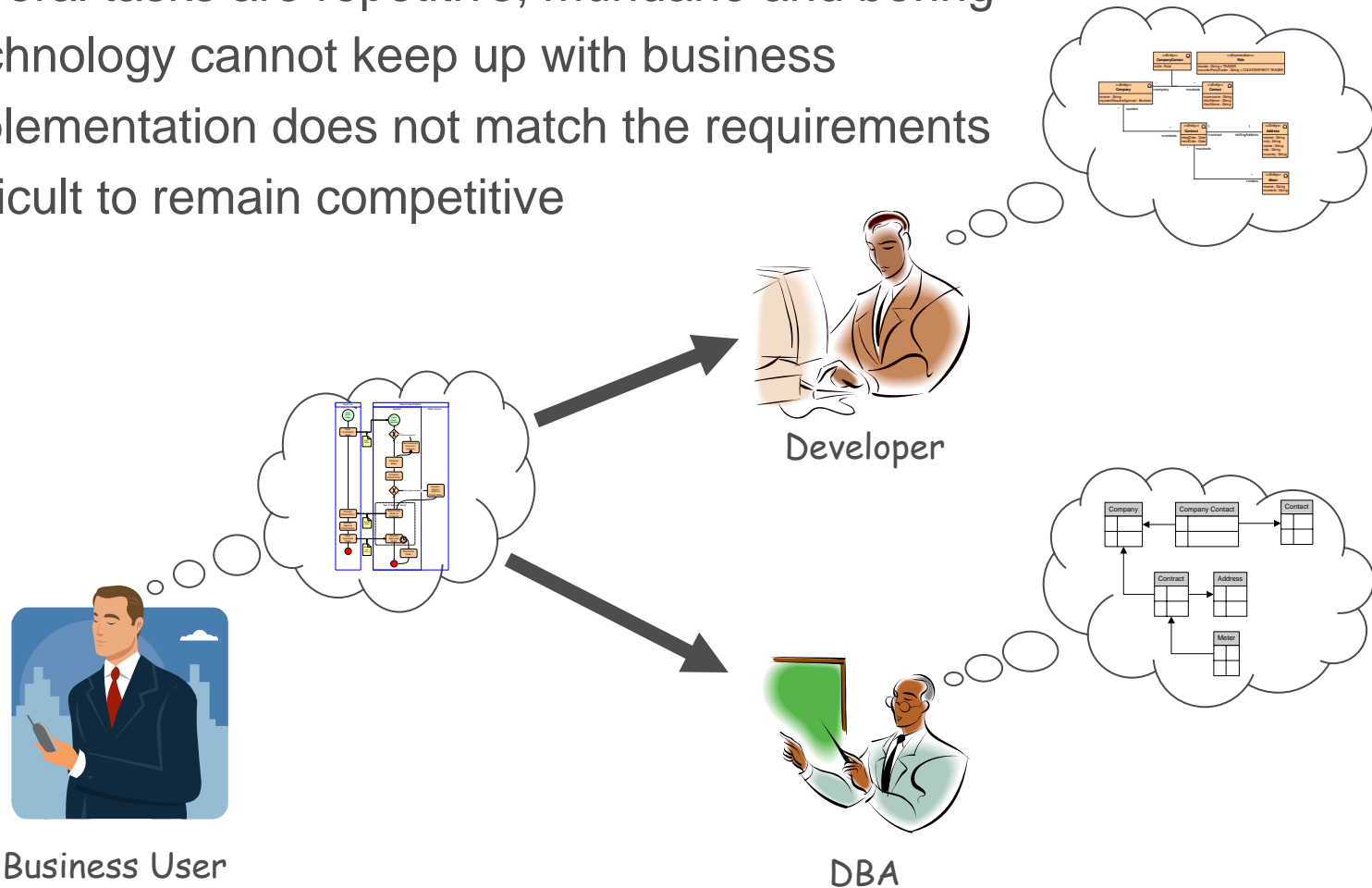
Translate these models to code.
Code a lot!



Test, fix and tune the “crude” application until it is refined.
Iterate, iterate, iterate...

What's wrong with this approach?

- Laborious and inefficient
- Several tasks are repetitive, mundane and boring
- Technology cannot keep up with business
- Implementation does not match the requirements
- Difficult to remain competitive

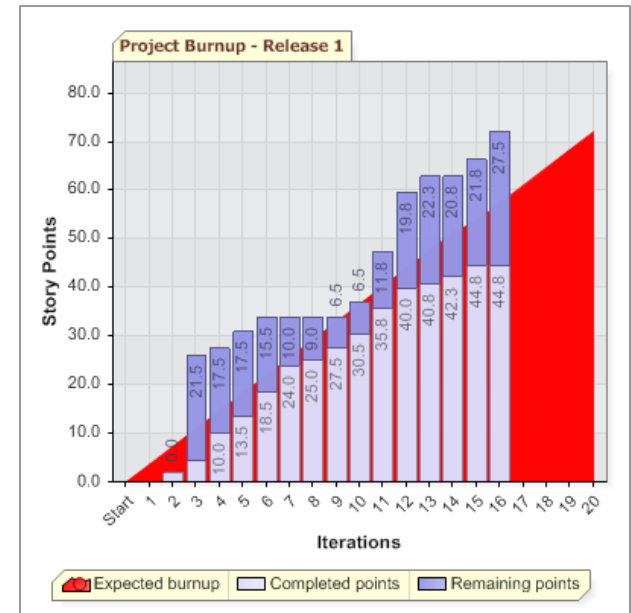
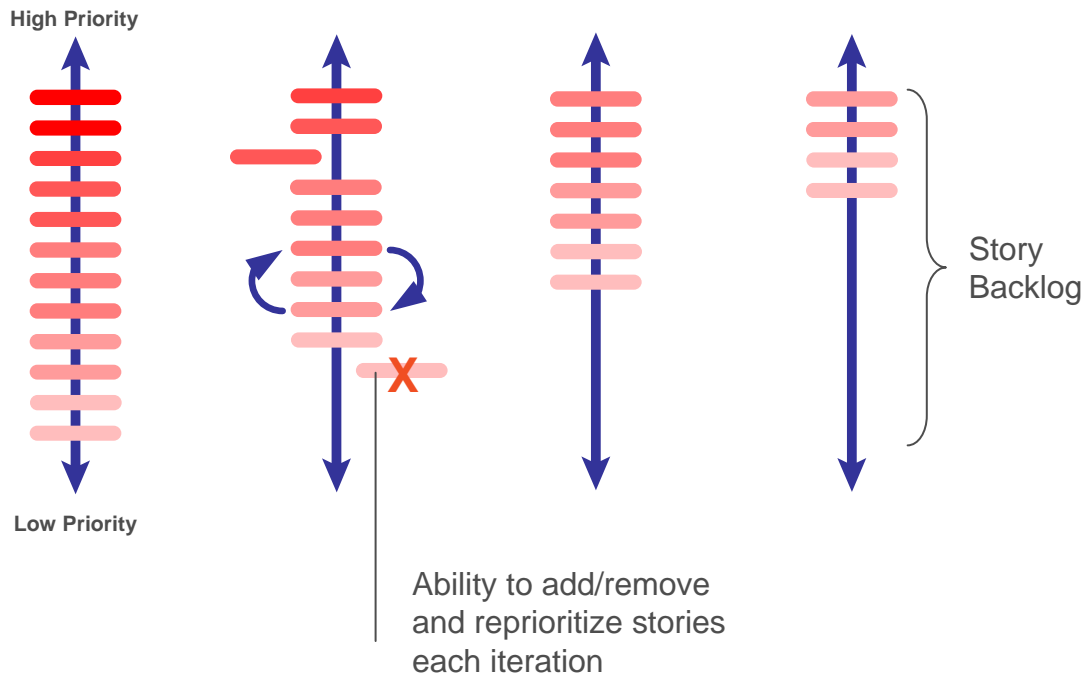
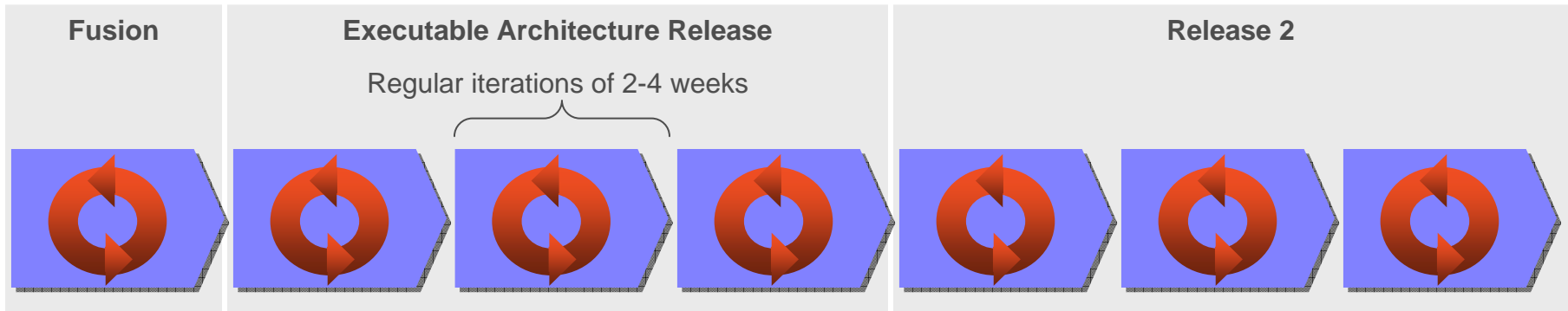


What can we do to change this?

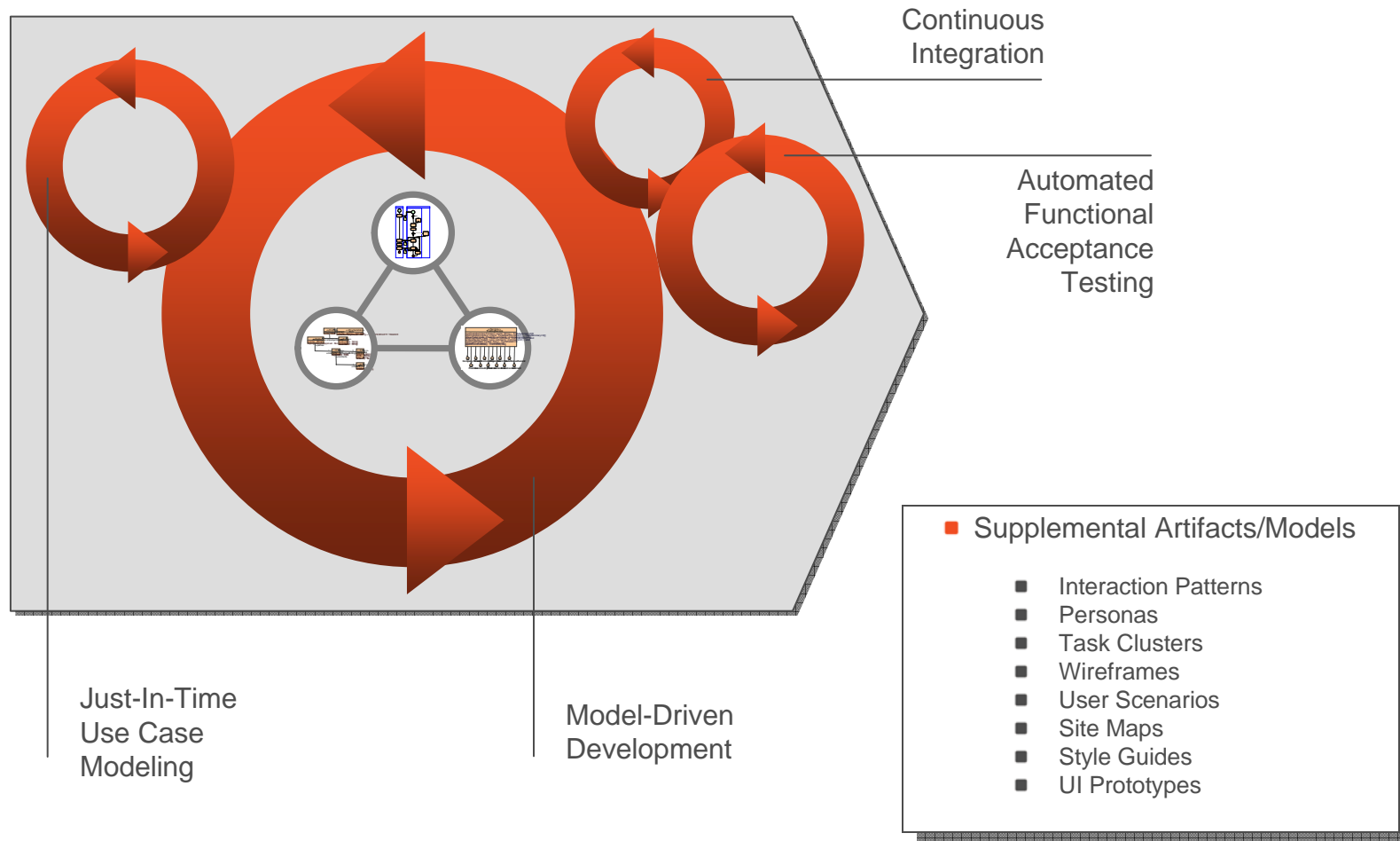
- Create a process that adapts to changing business needs
 - Provide frequent opportunities to business/end users to offer feedback
- Raise the level of abstraction
 - Let developers focus on the business problem – not on technology trivia
- Eliminate manual translations to get from requirements to code
 - Recognise repetitive tasks and patterns
 - Automate, automate, automate!

So how do we do that?

Introducing Sapien|Approach



Sapient|Approach – A typical iteration



Model Driven Development

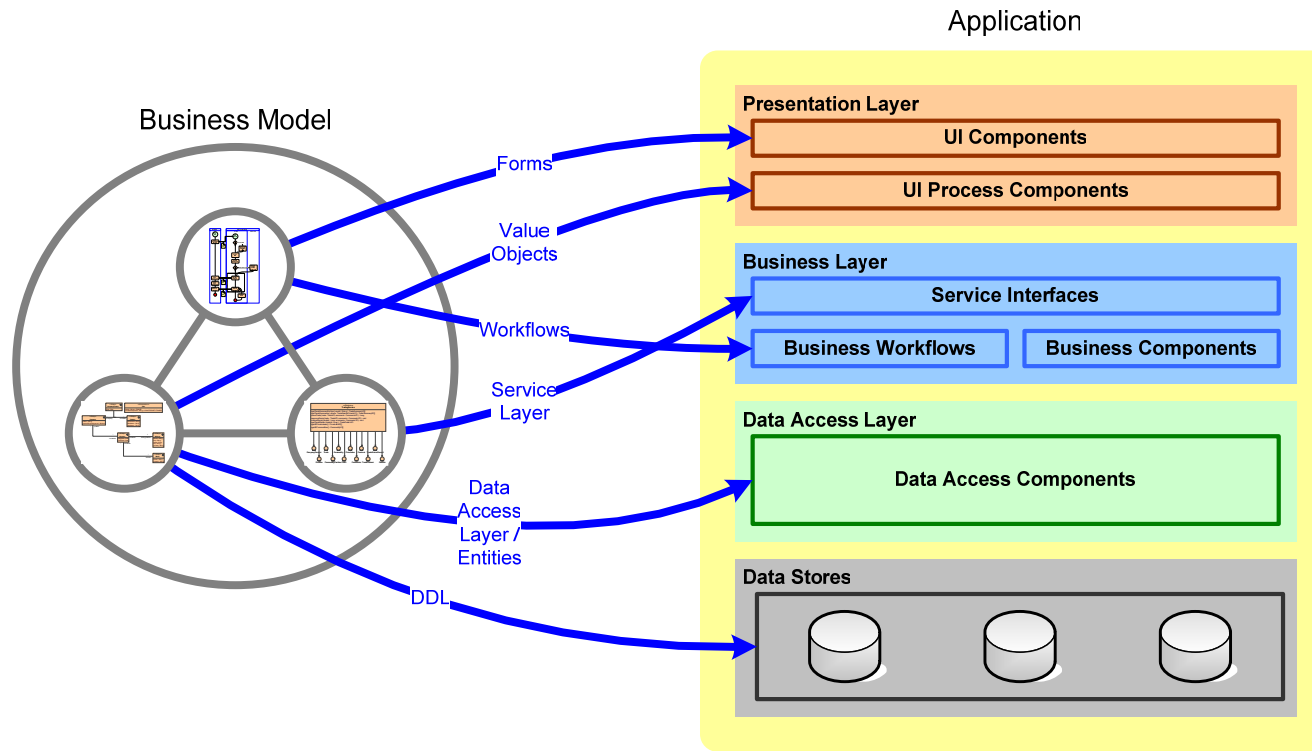
- A software development approach where system implementation is **truly driven by the business model**
- The business model is **independent of technology**
- Business model is automatically translated into a technology solution by identifying **common patterns**
- Business concepts are **described only once** but drive multiple application layers and multiple technologies
- Supports **system evolution** as technologies come and go
- Brings the fun back to application development!

NO FLUFF

Now That's Lean!™



So how does MDD work?



- All application layers driven from a single **business** model
- Business model automatically translated to technology model(s) and code
 - This is done by recognising common business patterns
- Example – a business entity can generate
 - A Plain Old Java (POJO)/C# Object
 - A Data Access Object (DAO)
 - SQL for generating database tables, constraints and indexes

But we already do that with CASE tools – what's different?

- Traditionally CASE tools have been used for
 - Business modelling and/or technology modelling
 - No connection between the two
- Code generation is usually done from technical models
 - One-to-one mapping between classes in the model and classes in code
- Model Driven Development changes all that!
 - Business model generates entire layers of your application
 - One model element can generate several implementation artefacts
- What are the enablers that make this possible now?
 - Standardisation of service models and technologies such as
 - Web Services
 - J2EE and .NET platforms
 - Availability of innovative application frameworks such as
 - Hibernate / NHibernate
 - Spring / Spring.NET
 - Struts / ASP.NET
 - Swing / Windows Forms
 - Advent of business-process centric platforms such as
 - jBPM / WebLogic Integration / WebSphere BI / Windows Workflow / BizTalk

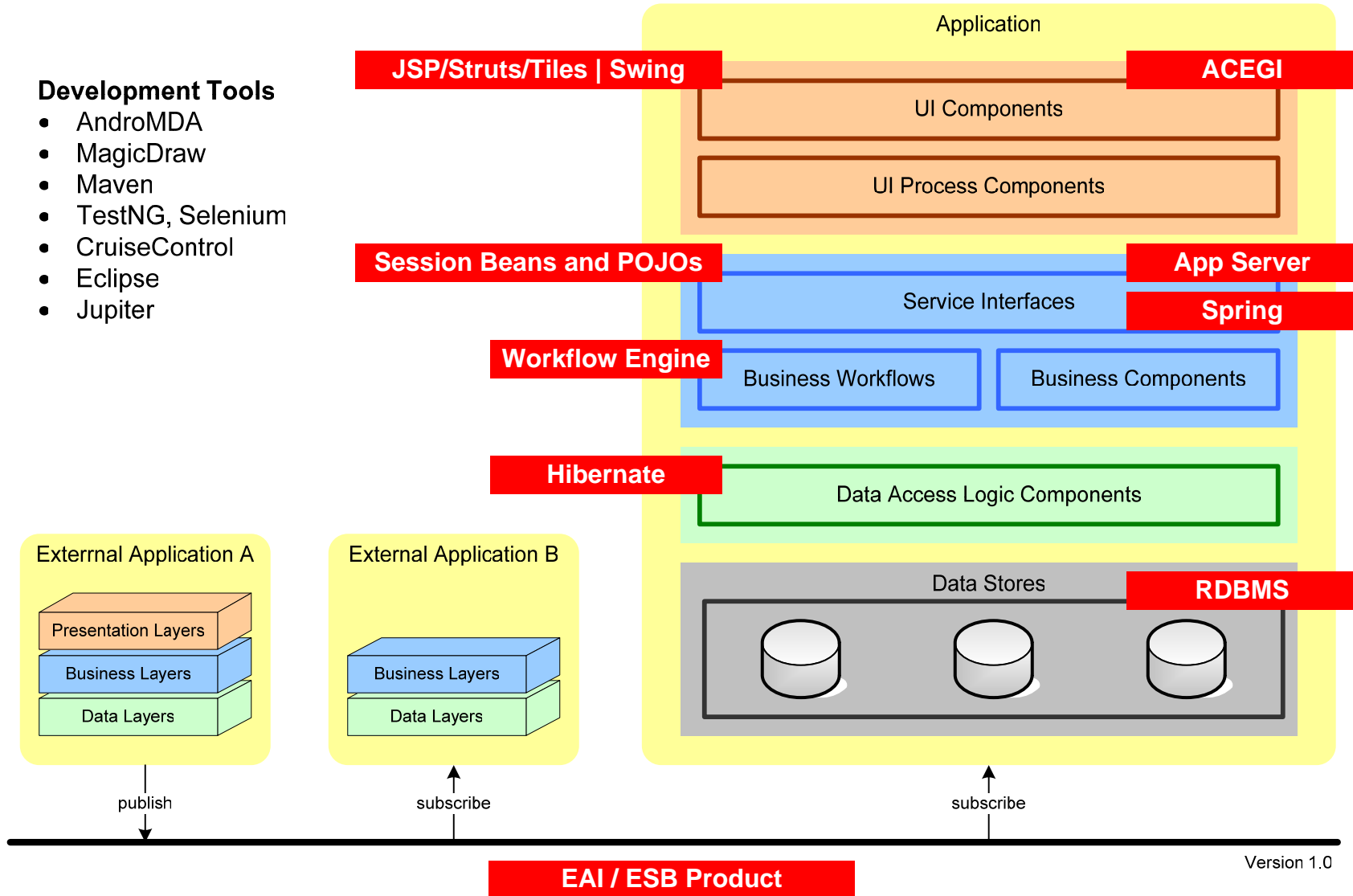
Sapient|Approach Technology Stack



S|A Technology Stack for Java

Development Tools

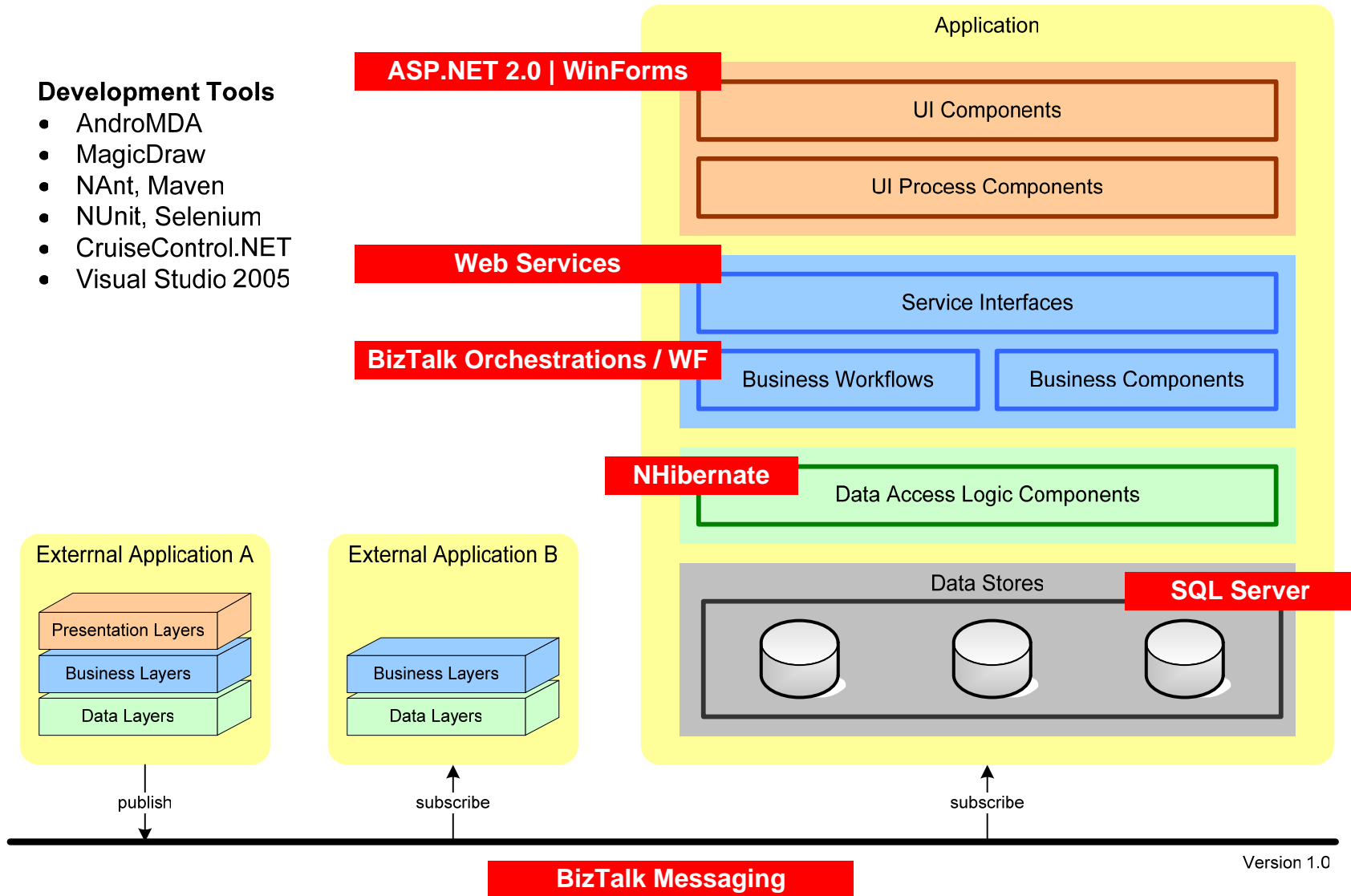
- AndroMDA
- MagicDraw
- Maven
- TestNG, Selenium
- CruiseControl
- Eclipse
- Jupiter



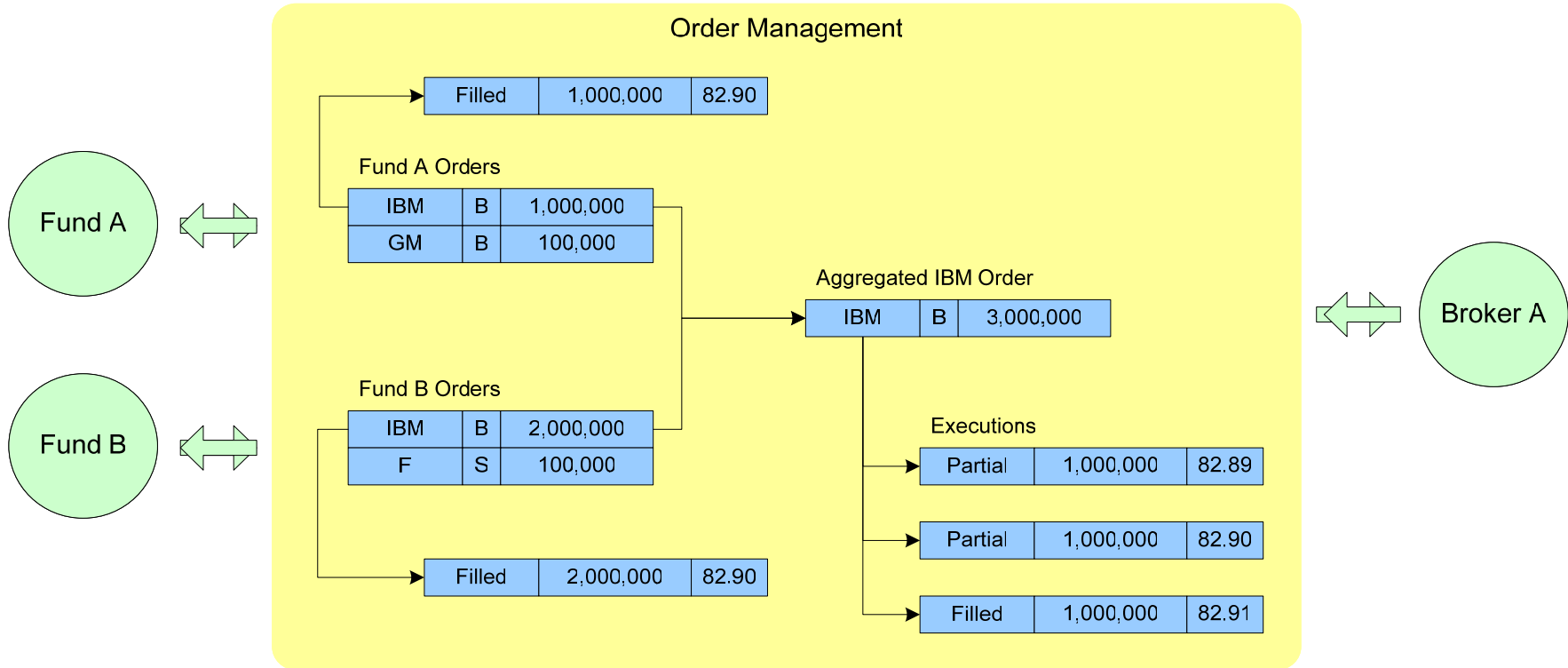
S|A Technology Stack for .NET

Development Tools

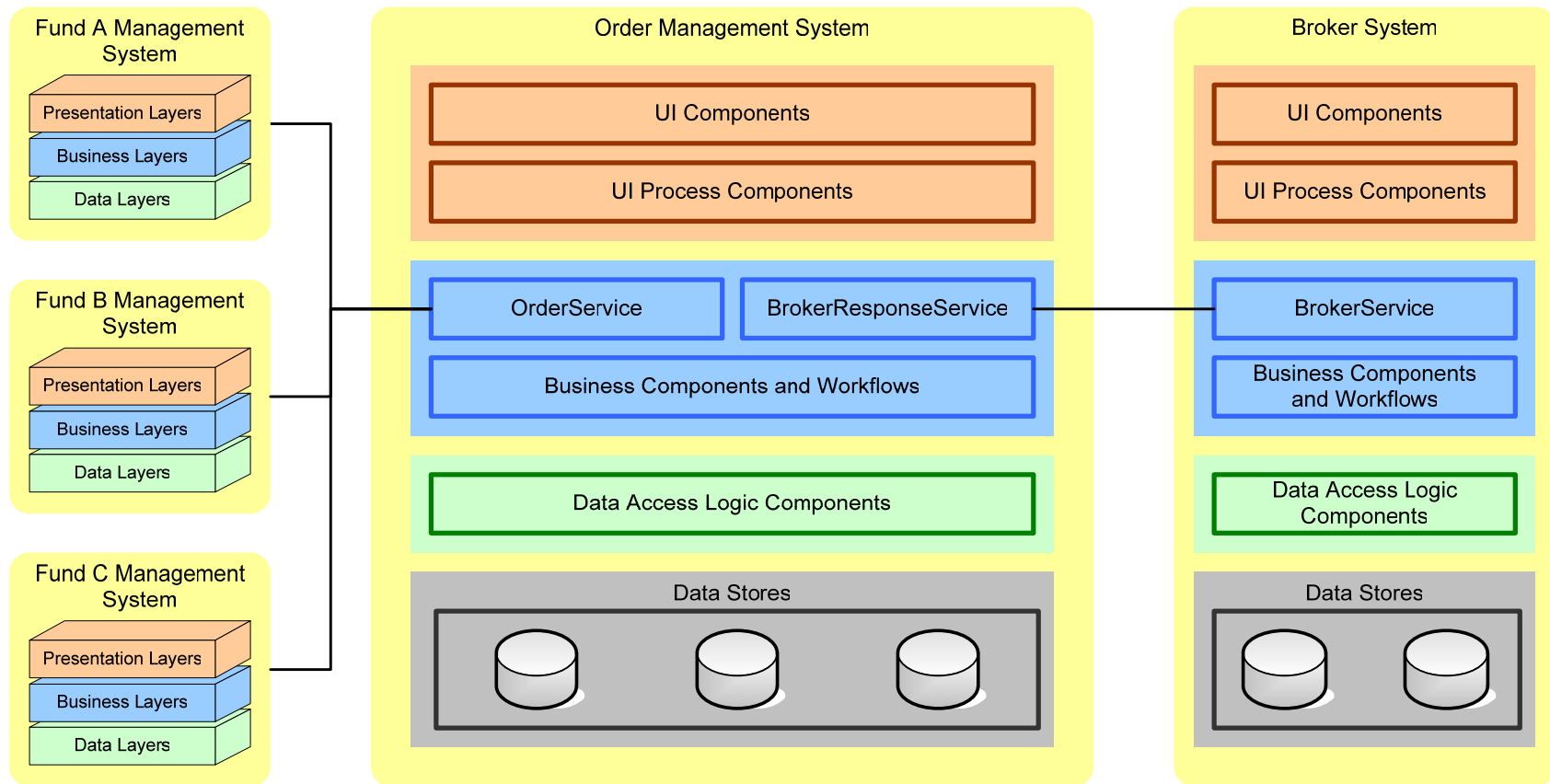
- AndroMDA
- MagicDraw
- NAnt, Maven
- NUnit, Selenium
- CruiseControl.NET
- Visual Studio 2005



Order Execution Scenario



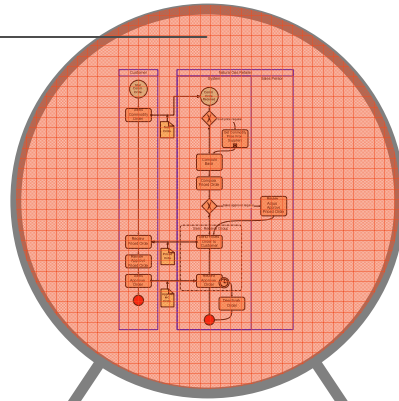
Logical Architecture



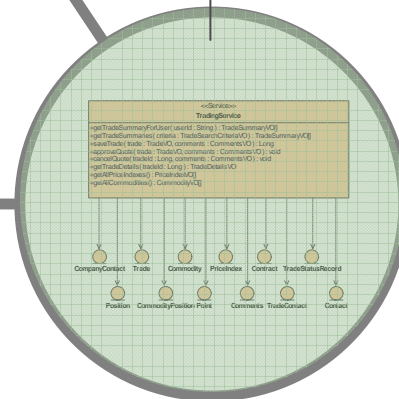
Key Elements of the Business Model

Process Model

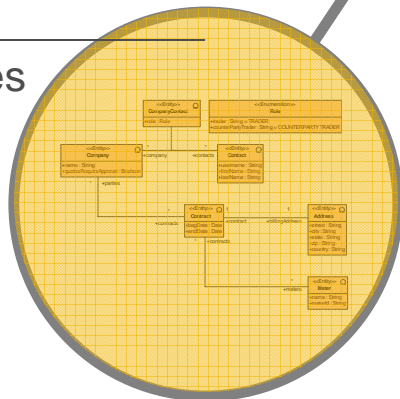
Use cases and business processes



Application behaviour



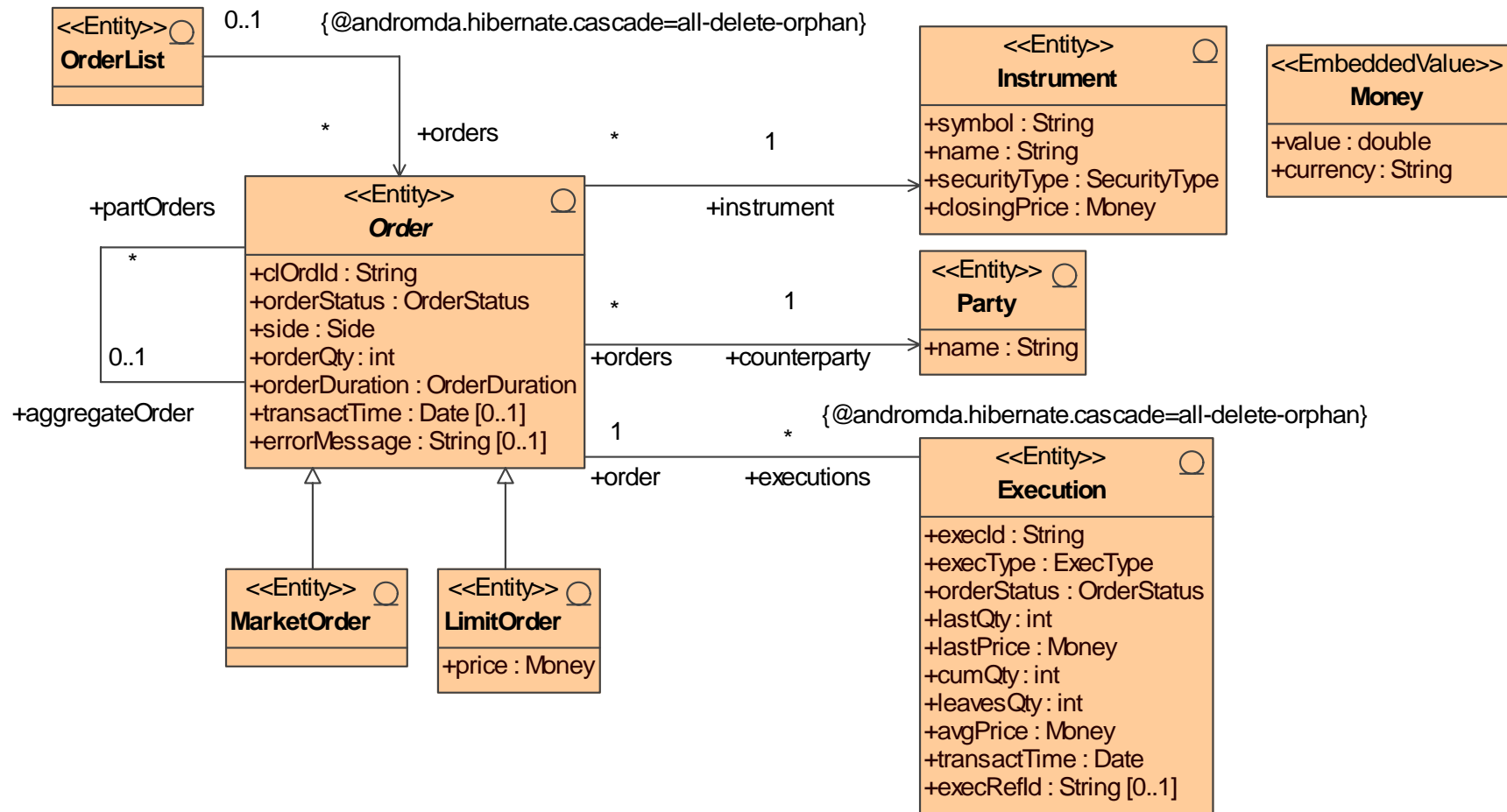
Business entities and their relationships



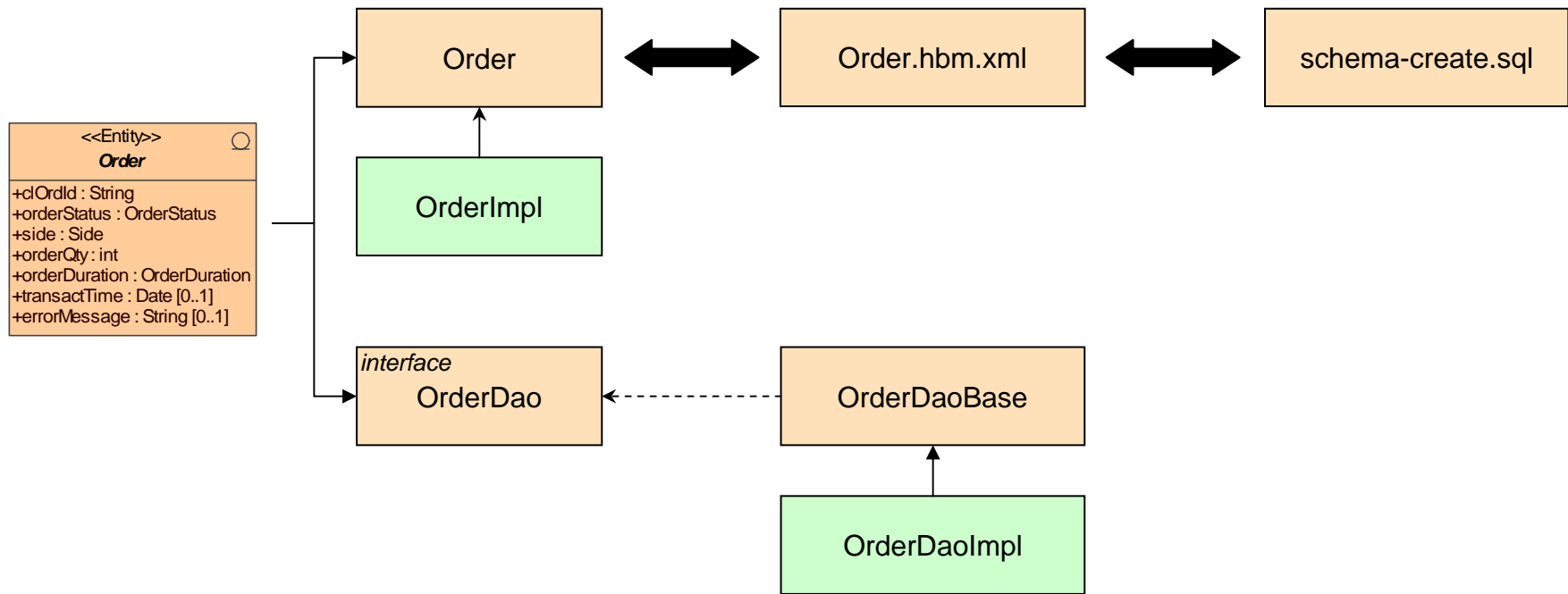
Object Model

Service Model

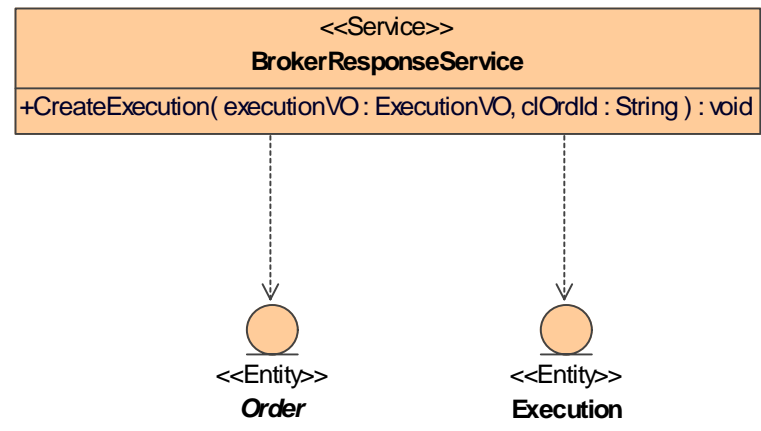
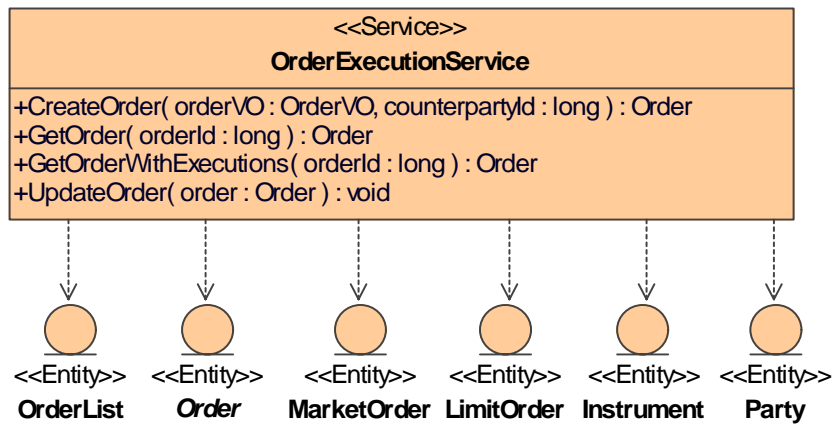
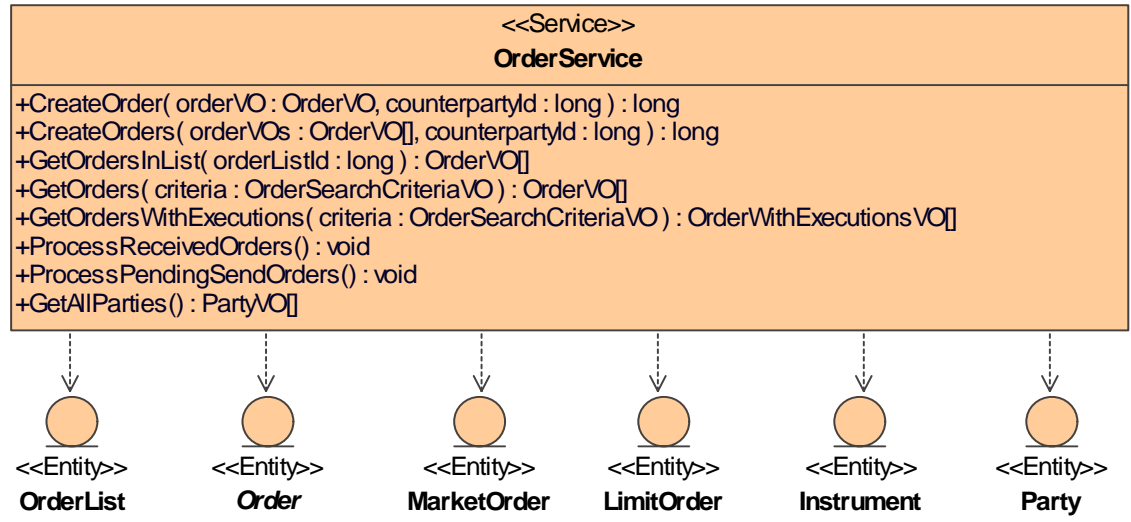
Object Model



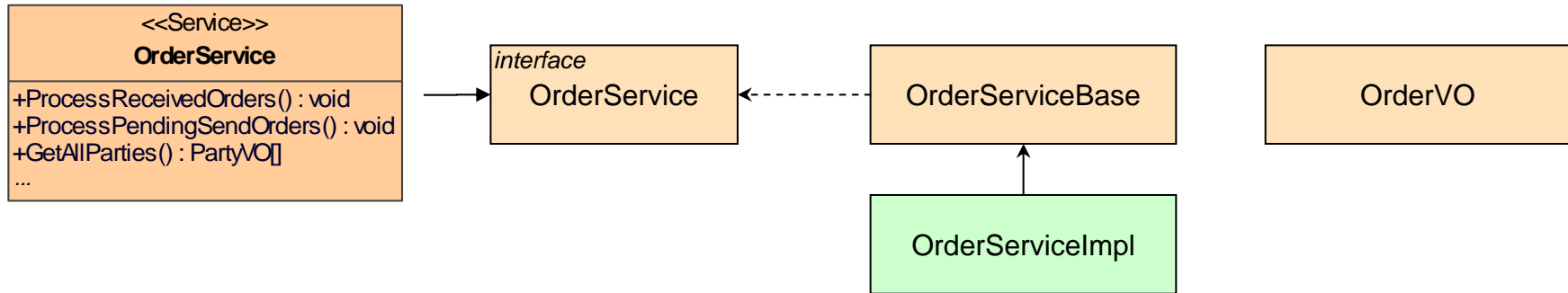
Code Generated for Order Entity



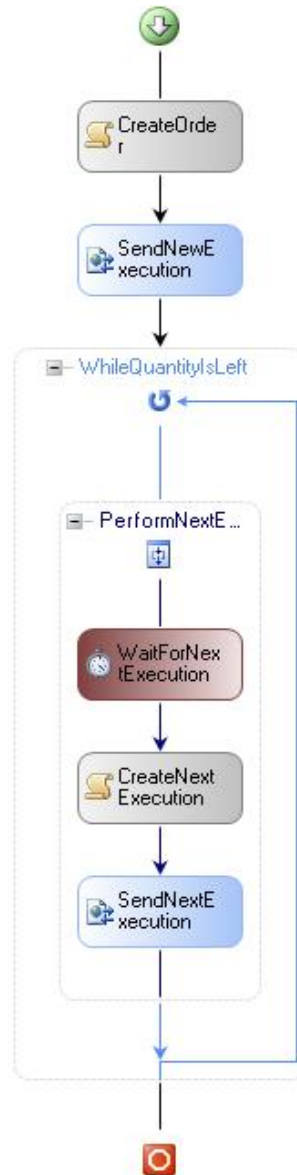
Service Model



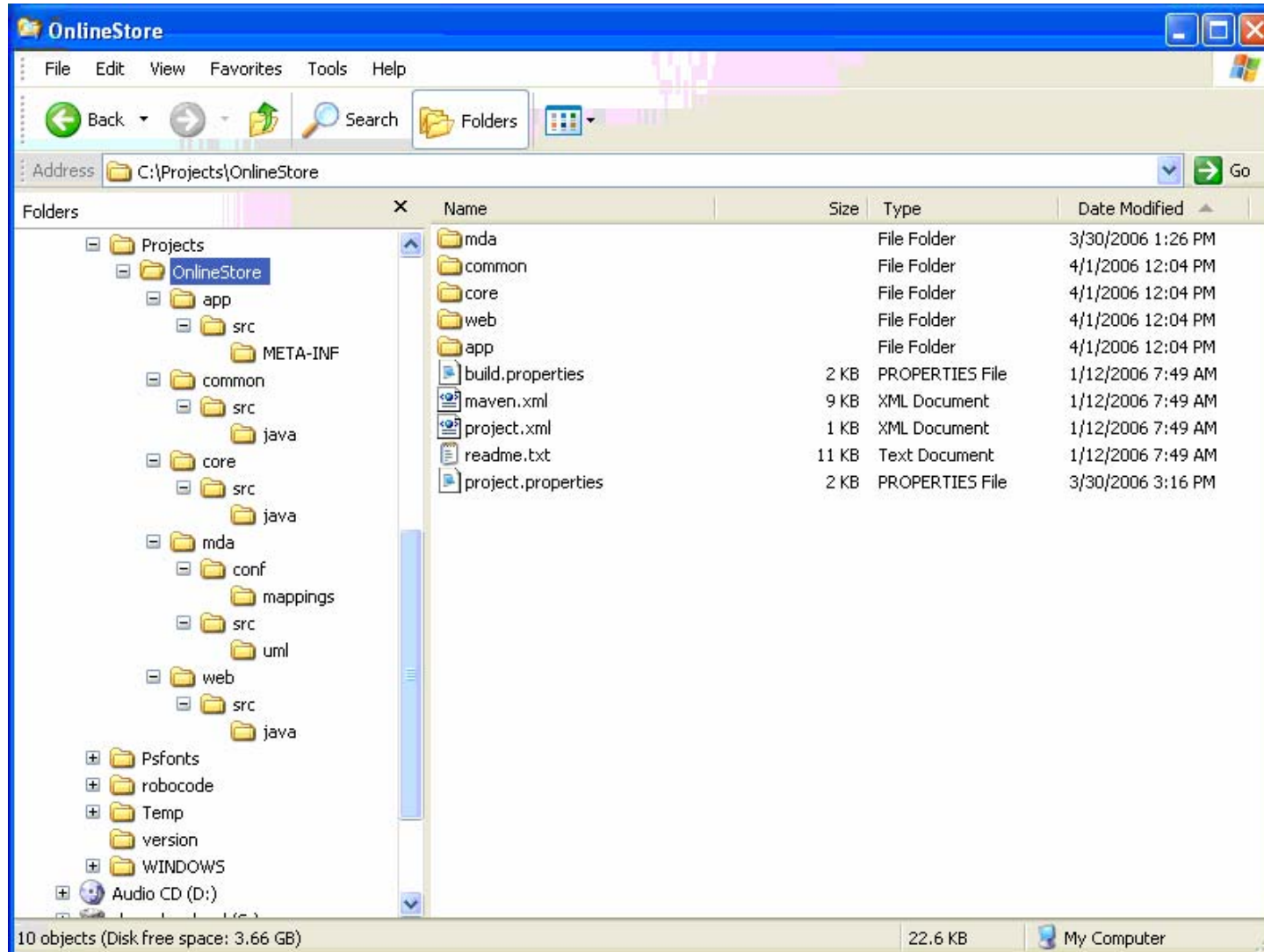
Code Generated for OrderService



Process Model



Demonstration



Pilot Projects, Results and Benefits



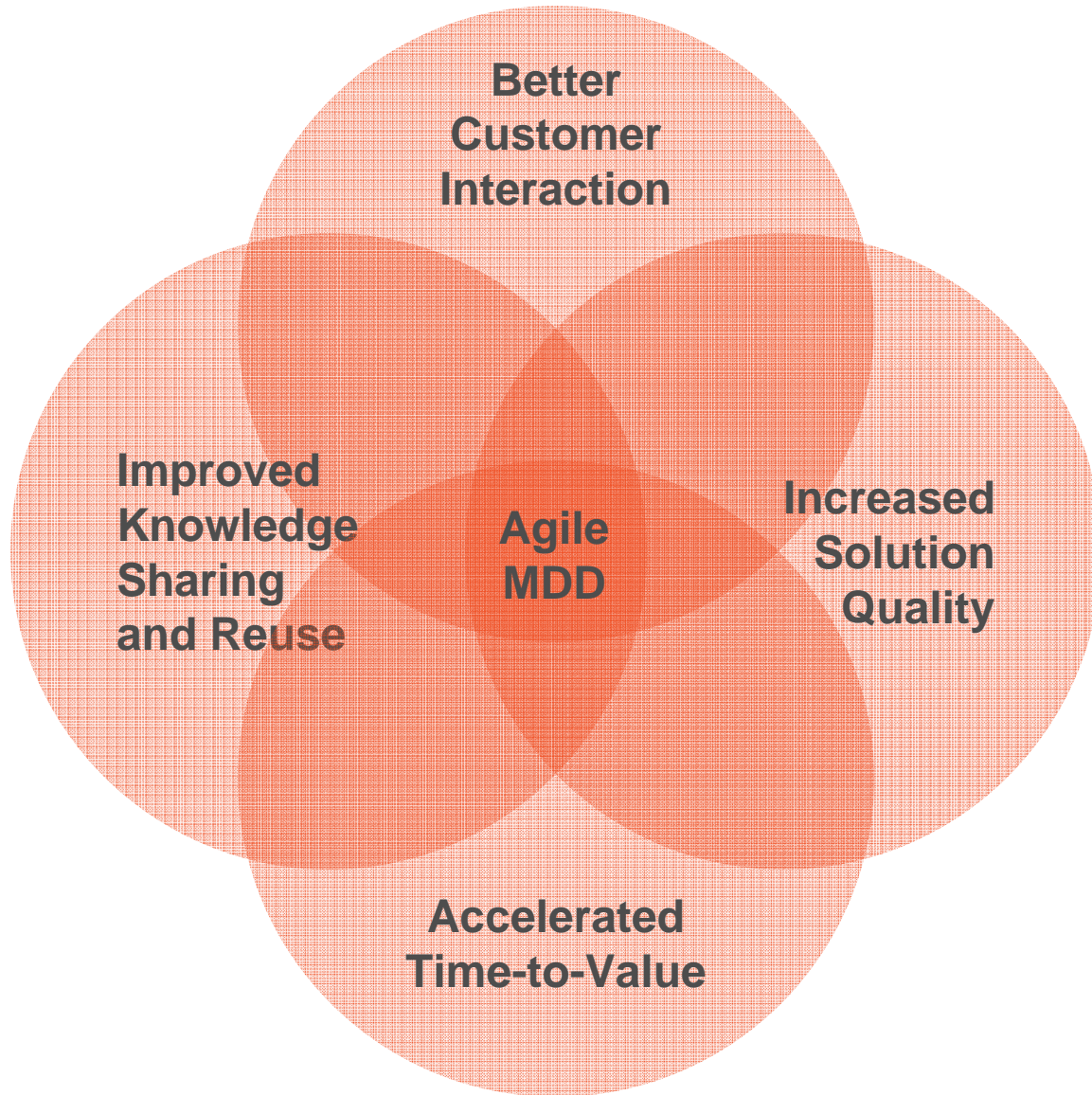
Pilot Projects

- High Energy
 - Reference application for the S|A Technology Stack
 - Increases operational efficiency in natural gas trading
 - Achieved 25% LOE reduction over traditional approach
- German Telecom client
 - Value-added services based on directory data
 - Delivered in 6 weeks, although estimates using traditional techniques predicted this was not possible
- North American Energy Services client
 - Project 1: Lower natural gas costs by aggregating customer needs
 - Project 2: Collection of gas usage information from disparate sources
 - Reduced overall development time and effort, higher quality results
- North American Financial Services client
 - Replaced EJB2 style entity beans with Hibernate 3.1
 - The targeted use case of processing 250 items dropped from 31 sec. to around 15 sec. With L2 cache, it dropped to under 10 sec. The desired SLA is 8 sec.
 - Approximately 50% of the code remains unchanged, which accounts for 7 sec. of the total time.
 - The amount of code, and code complexity was also significantly reduced

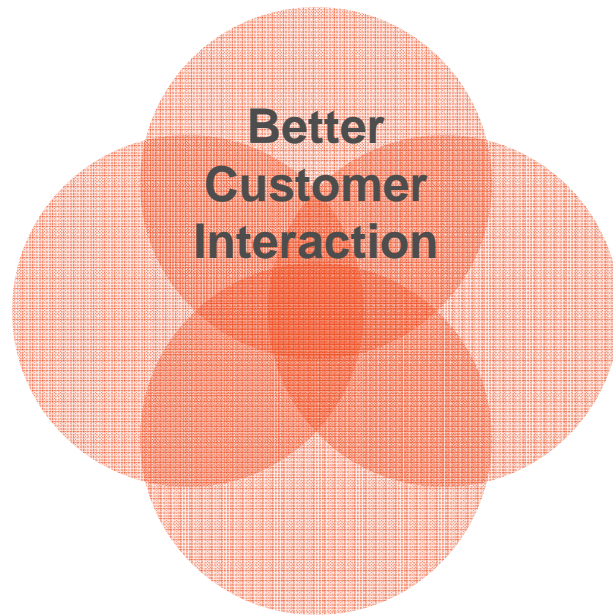
Results from High Energy

- Statistics: Middle-Tier
 - 85% of code generated automatically
- Statistics: Front-End
 - 68% of code generated automatically
- We believe that MDD provides at least 30% gain in productivity
 - Higher if team is highly experienced with MDD and the technology stack
- Similar results obtained by an independent study conducted by The Middleware Company
 - “The result of this study is the MDA team developed their application 35% faster than the traditional team.”
 - http://www.omg.org/mda/mda_files/MDA_Comparison-TMC_final.pdf

Benefits Achieved through Agile MDD

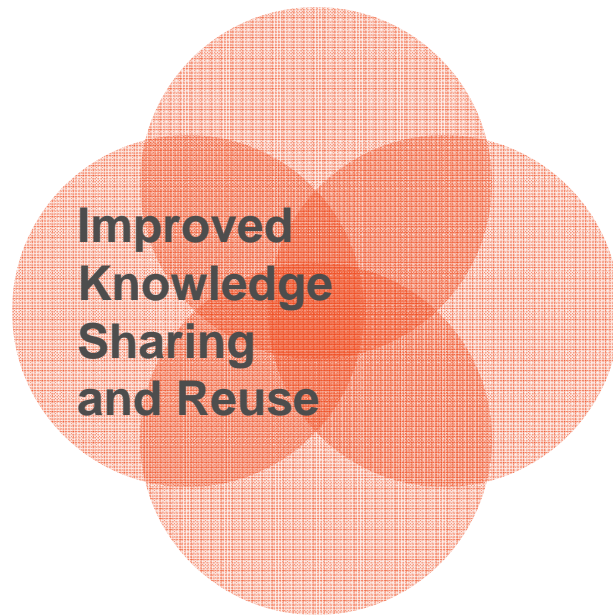


Benefits Achieved through Agile MDD



- Common language for business users and developers
 - Models describe business requirements precisely and succinctly
- Greater focus on getting the business model right
 - Code is just a by-product of well-understood customer requirements
- Models always represent the current state of the system
 - Not just a piece of outdated documentation
 - Models are first-class artefacts, as is source code
 - Solution's expected behaviour is unambiguous to all stakeholders

Benefits Achieved through Agile MDD



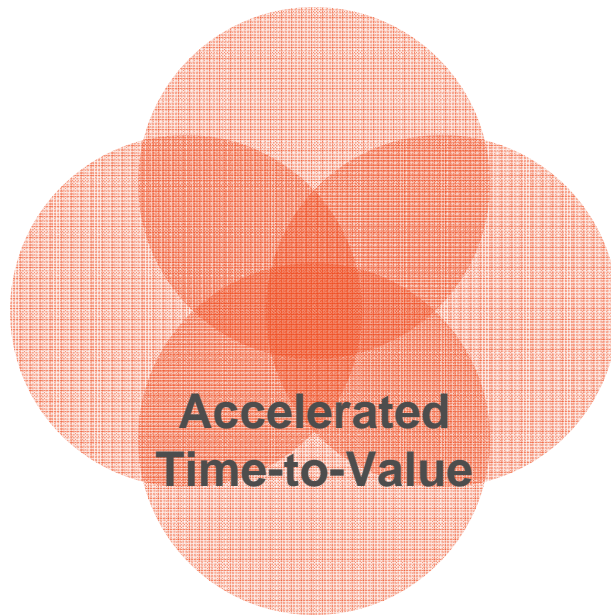
- Preserves investment in business models
 - Supports system evolution as technologies come and go
 - Enables expression of accumulated industry knowledge in the form of reusable domain-specific business models

Benefits Achieved through Agile MDD



- More thought put into design patterns instead of hacking code
 - Relentless commitment to simple design and YAGNI
- Reduces possibility of human error

Benefits Achieved through Agile MDD



- Lower level of effort
 - Finally, there is a way to automatically crank out mundane code
- Faster response to changing business needs
 - Developers are less likely to resist requirement changes
 - Code generation takes care of low-level implementation details



Questions and Feedback



Agile India, 5-6 May, 2006, Bangalore, India
egottesman@sapient.com