

# Successful Enterprise Architecture

Ivar Jacobson

[ivar@ivarjacobson.com](mailto:ivar@ivarjacobson.com)

[ivar@jaczone.com](mailto:ivar@jaczone.com)

---

**Ivar Jacobson International**

# Motivation

Life is becoming increasingly more complex

- Business is more complex
- Software is becoming more complex

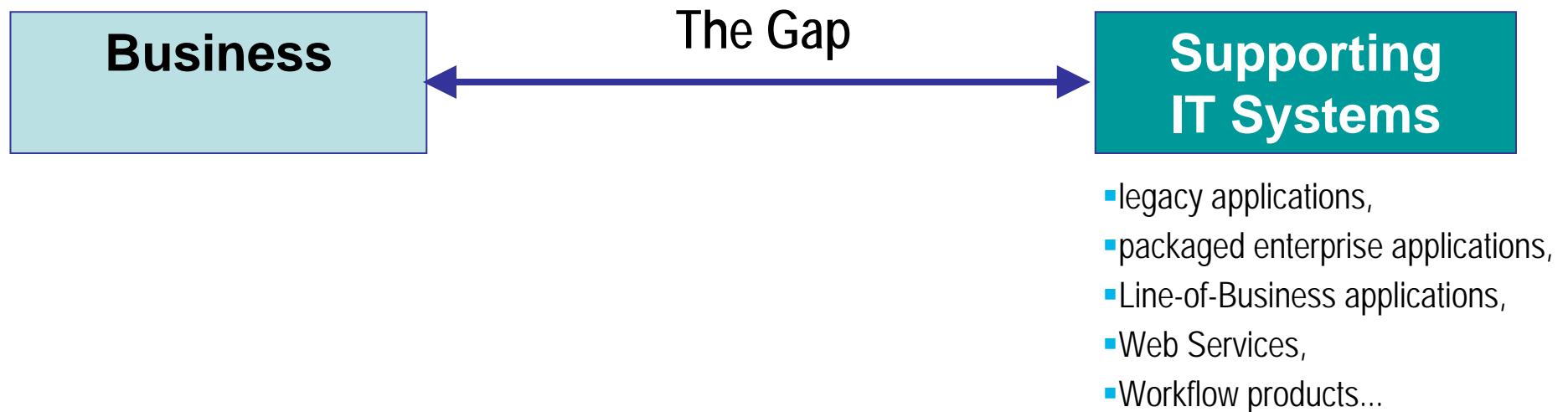


# Agenda

- Gaps in enterprise IT
- Realities in enterprise IT
- Need for enterprise architecture
- Ingredients for a good enterprise architecture
- Our approach towards enterprise architecture
- Applying our approach in practice
- Turn enterprise architecture to reality

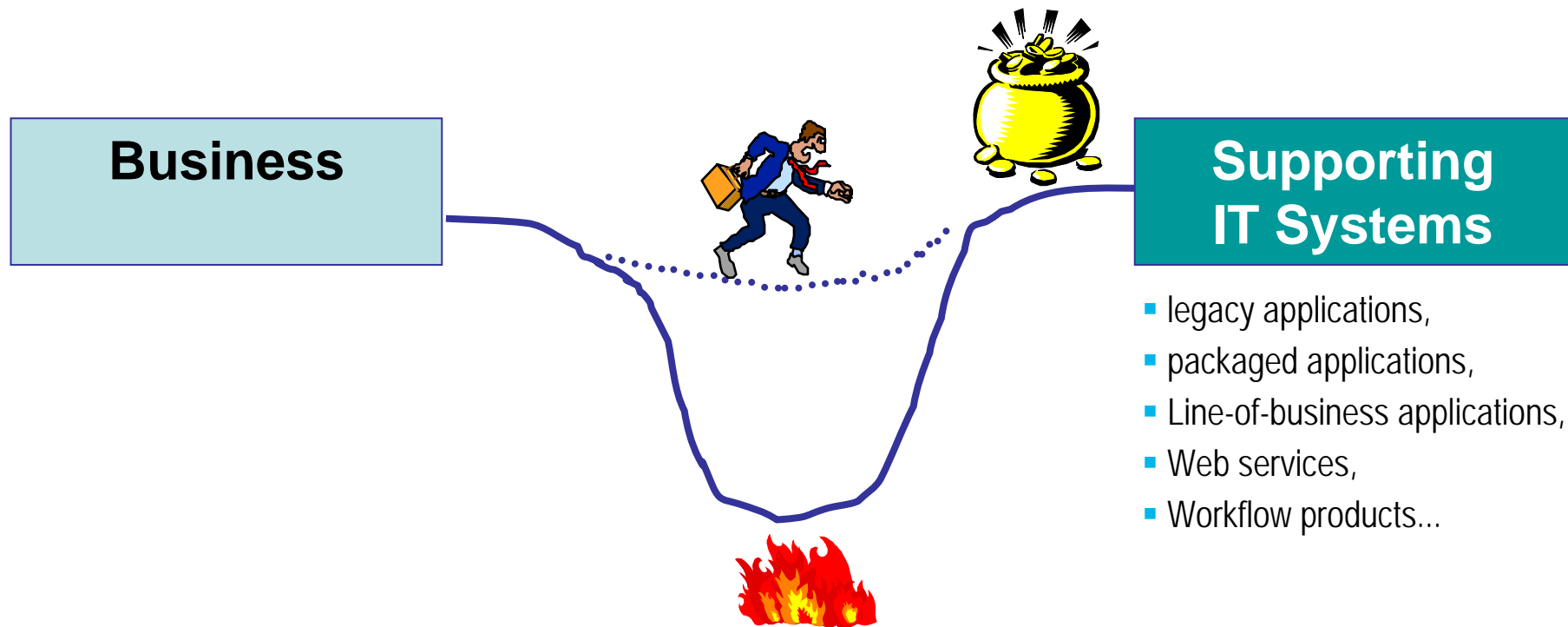
# Gap between business and IT

- The problem is the gap between business requirements and the systems that supports the business

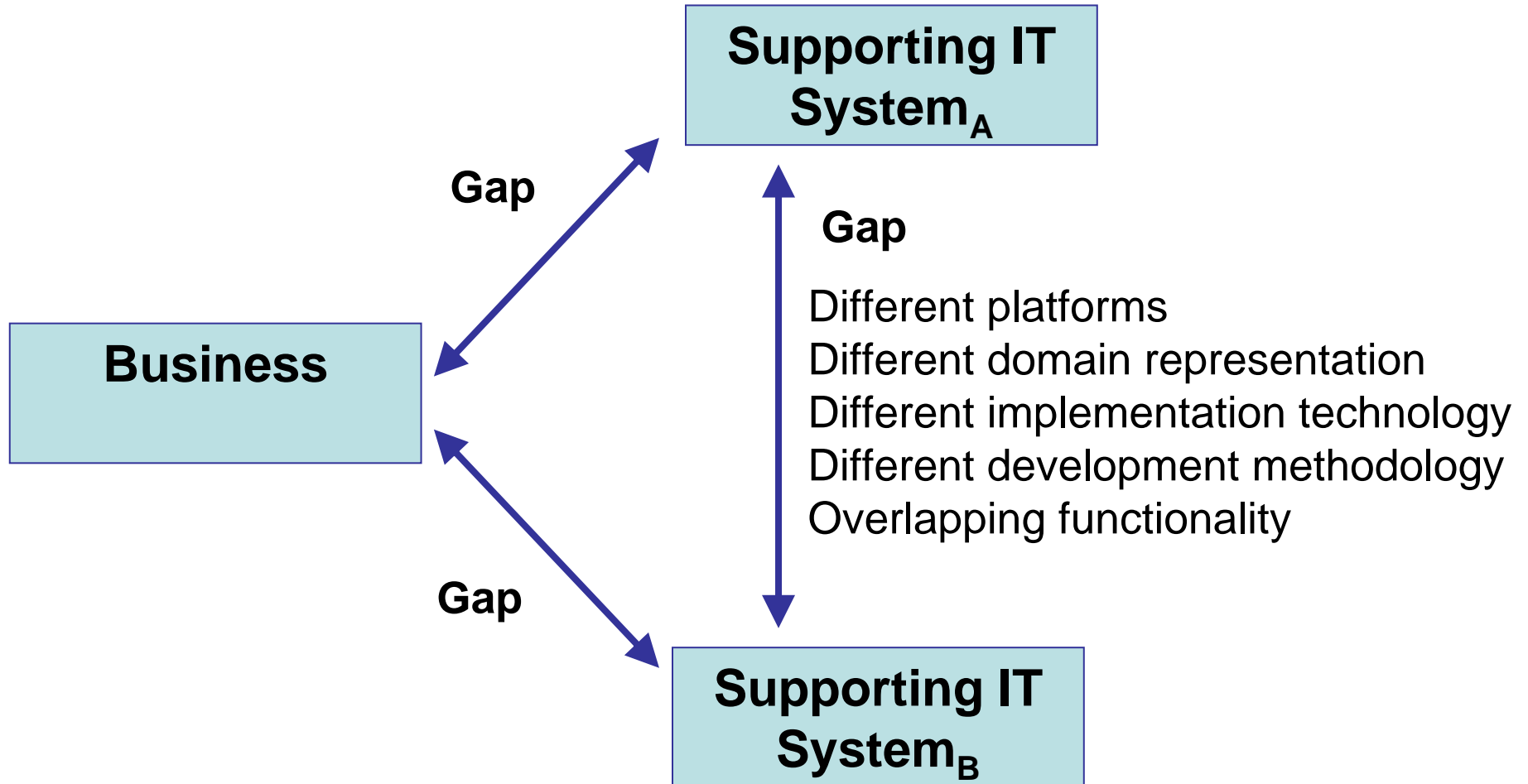


# Gap between business and IT

- The problem is the gap between business requirements and the systems that supports the business
- We need to close the gap!



# There are many gaps to close today

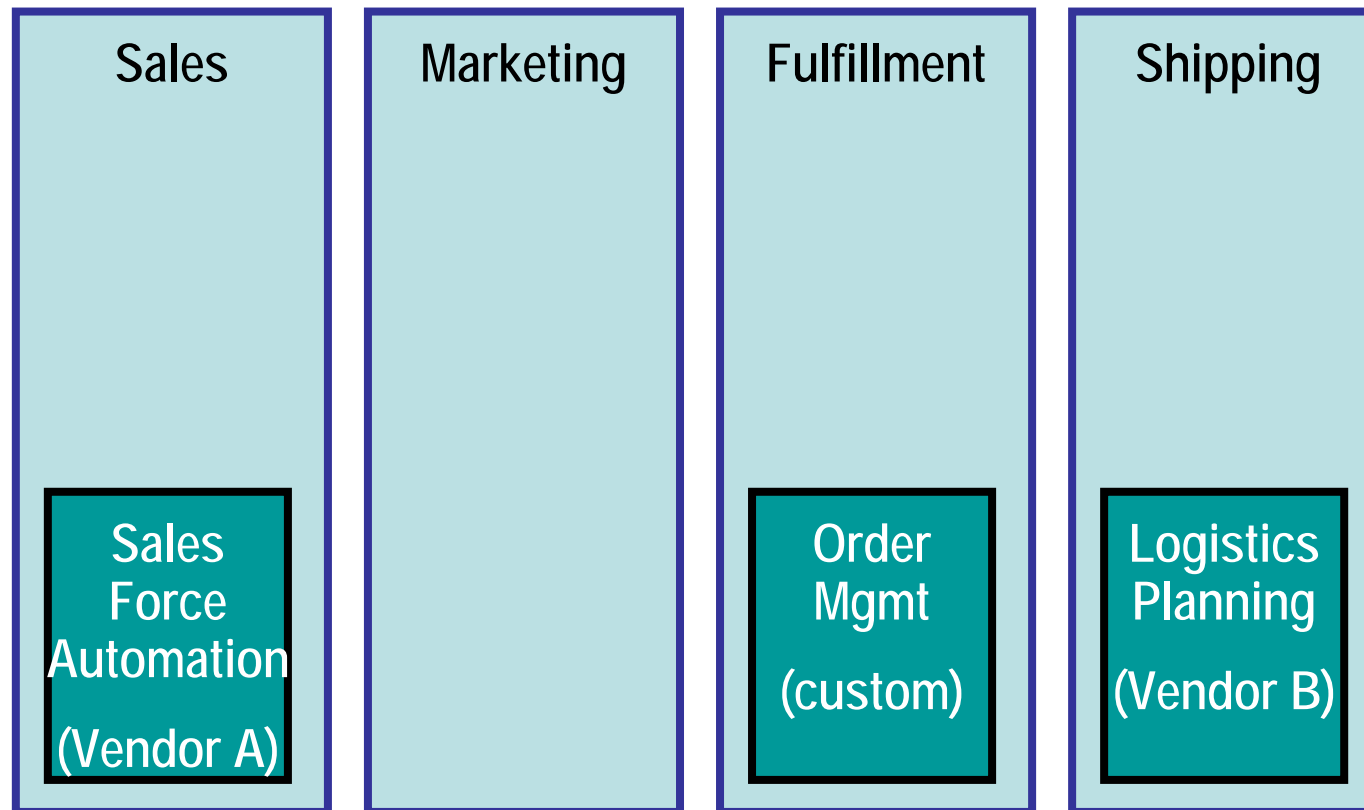


# Agenda

- Gaps in enterprise IT
- Realities in enterprise IT
- Need for enterprise architecture
- Ingredients for a good enterprise architecture
- Our approach towards enterprise architecture
- Applying our approach in practice
- Turn enterprise architecture to reality

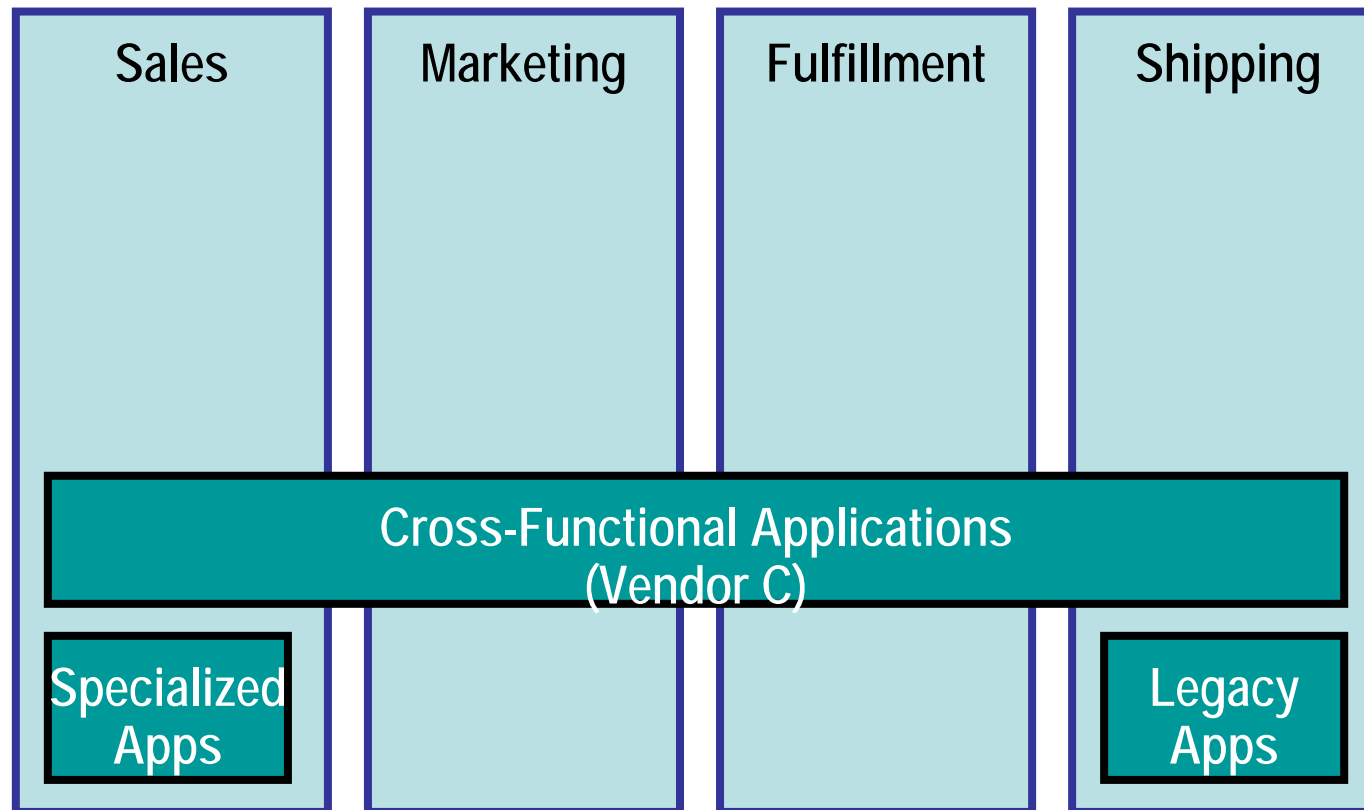
# The realities of enterprise IT

- Most businesses started with custom developed, stove-pipe, functional applications



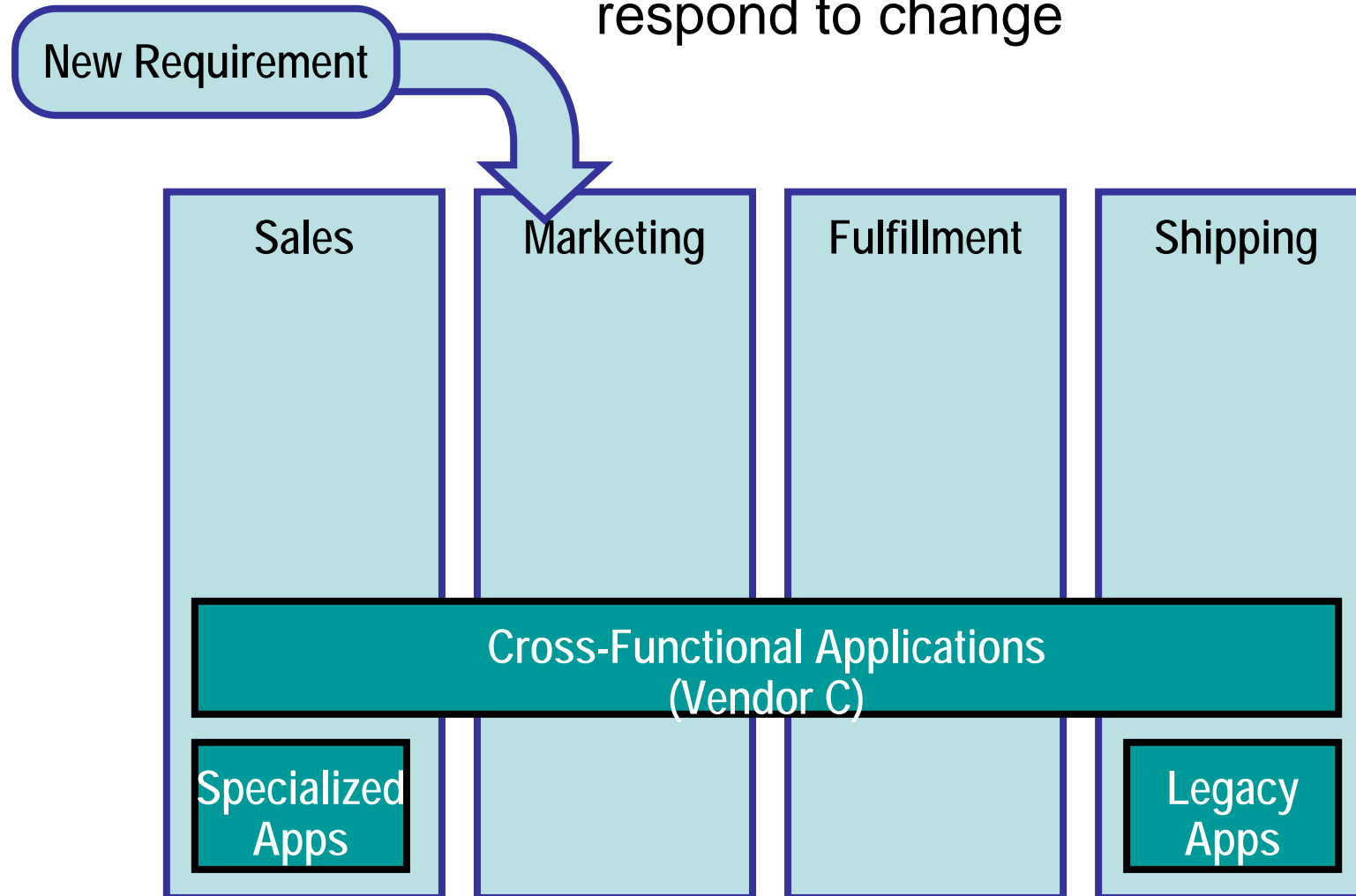
# The realities of enterprise IT

- Recent moves to packaged enterprise applications have helped reduce the cost of maintaining legacy applications



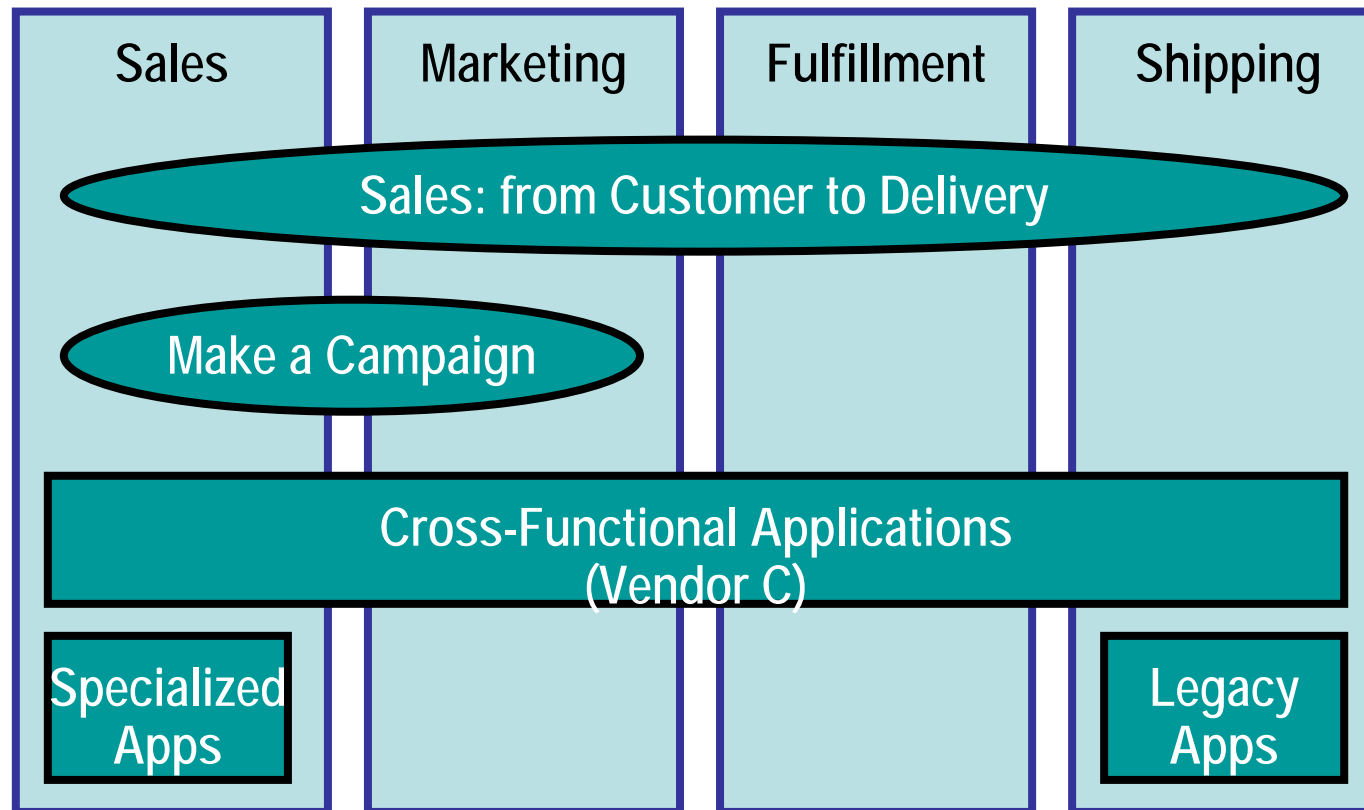
# The realities of enterprise IT

- However enterprise applications are still slow to respond to change



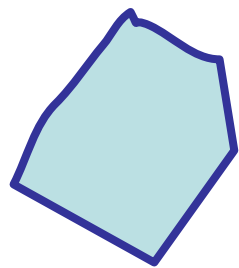
# The realities of enterprise IT

- And frequently do not support cross-organizational processes well

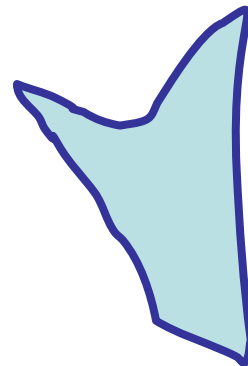


# Challenges with existing applications

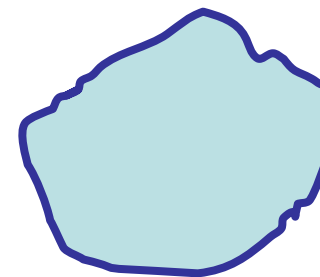
- They all "look" different. "Homegrown" approach to describe them – no standard
  - Hard to understand
  - Hard to reuse or extend
  - Hard to interconnect with other applications
  - Hard to replace with another vendor's applications



Enterprise  
Resource  
Planning  
(Vendor A)



Customer  
Relationship  
Management  
(Vendor B)



Legacy

## Challenges with existing applications

- They all "look" different. "Homegrown" approach to describe them – no standard
  - Hard to understand
  - Hard to reuse or extend
  - Hard to interconnect with other EA's
  - Hard to replace with another vendor's EA

**We will need to change that!**



Enterprise  
Resource  
Planning  
(Vendor A)

Customer  
Relationship  
Management  
(Vendor B)

Legacy

# Agenda

- Gaps in enterprise IT
- Realities in enterprise IT
- Need for enterprise architecture
- Ingredients for a good enterprise architecture
- Our approach towards enterprise architecture
- Applying our approach in practice
- Turn enterprise architecture to reality

# The need for enterprise architecture

- To realize the business processes by unifying available applications, domains, infrastructure, platforms
- However, this is not for free
  - You must understand the business process to be able to integrate applications
  - You must understand the applications to be able to integrate them

## The need for enterprise architecture

- To realize the business processes by unifying available applications, domains, infrastructure, platforms
- However, this is not free
  - You must create a strategy that is able to integrate applications
  - You must use a framework that is able to integrate them

Thus to understand  
is key!

# A common way to attack these problems are through modeling

## The value of models

- Understand the problem
- Communicate with stakeholders
- Help find risks
- Find errors and omissions
- Simulate complex situations
- Analyse changes

## The value of models

- Understand the problem
- Communicate with stakeholders
- Help find risks
- Find errors
- Simulate complex systems
- Analyze

**You will need to Model!**  
Thus to understand is key!

## Challenges with modeling as used today

- No standard business modeling language
  - Competing vendor standards have fragmented market
- No seamless transition from business modeling to system design and vice versa
  - System and business modeling are not integrated
- The business modeling process is tedious and error-prone
  - Leads to *"modeling paralysis"*
  - Keeping models up-to-date is expensive

## Challenges with modeling as used today

- No standard business modeling language
  - Competing vendor standards have fragmented market
- No seamless transition from business modeling to system design and vice versa
  - System and business modeling are not integrated
- The business modeling process is tedious and error-prone
  - Leads to "modeling paralysis"
  - Keeping models up-to-date is expensive

**We will need to change that!**

## Modeling is not enough

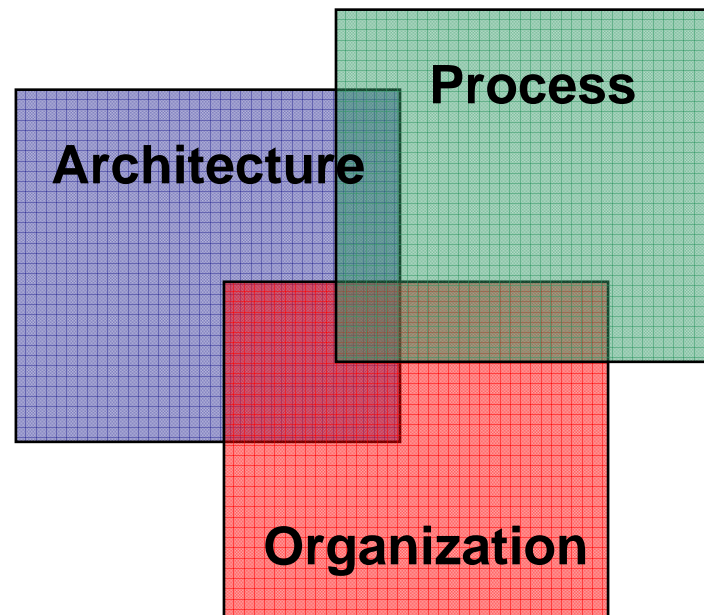
- Models help you describe what you have and what you want
- How do you bring models to reality?
  - You want real software running in your business
  - Not just models
- Enterprise architecture is not just about modeling
- Enterprise architecture also involves a process
- Enterprise architecture also involves the organization of the software development

# Agenda

- Gaps in enterprise IT
- Realities in enterprise IT
- Need for enterprise architecture
- Ingredients for a good enterprise architecture
- Our approach towards enterprise architecture
- Applying our approach in practice
- Turn enterprise architecture to reality

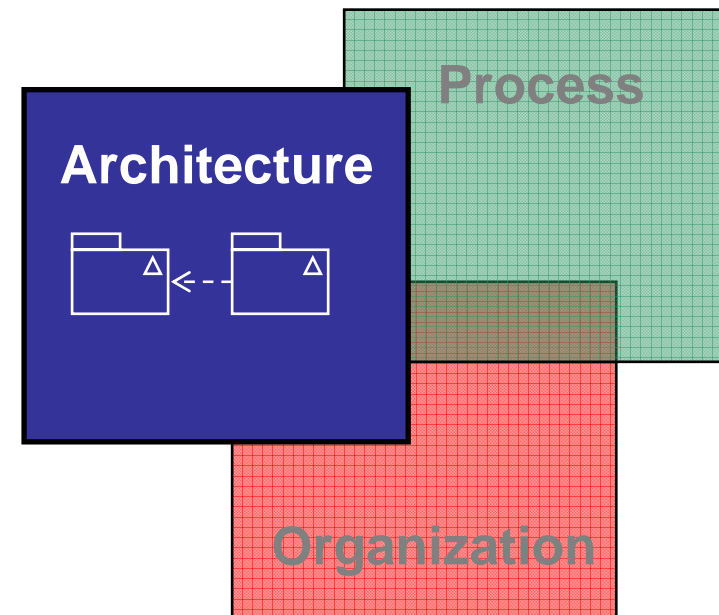
# What is enterprise architecture about?

- Architecture is about *what* must be done
- Process is about *how* and *when* it will be done
- Organization is about *who* will do it



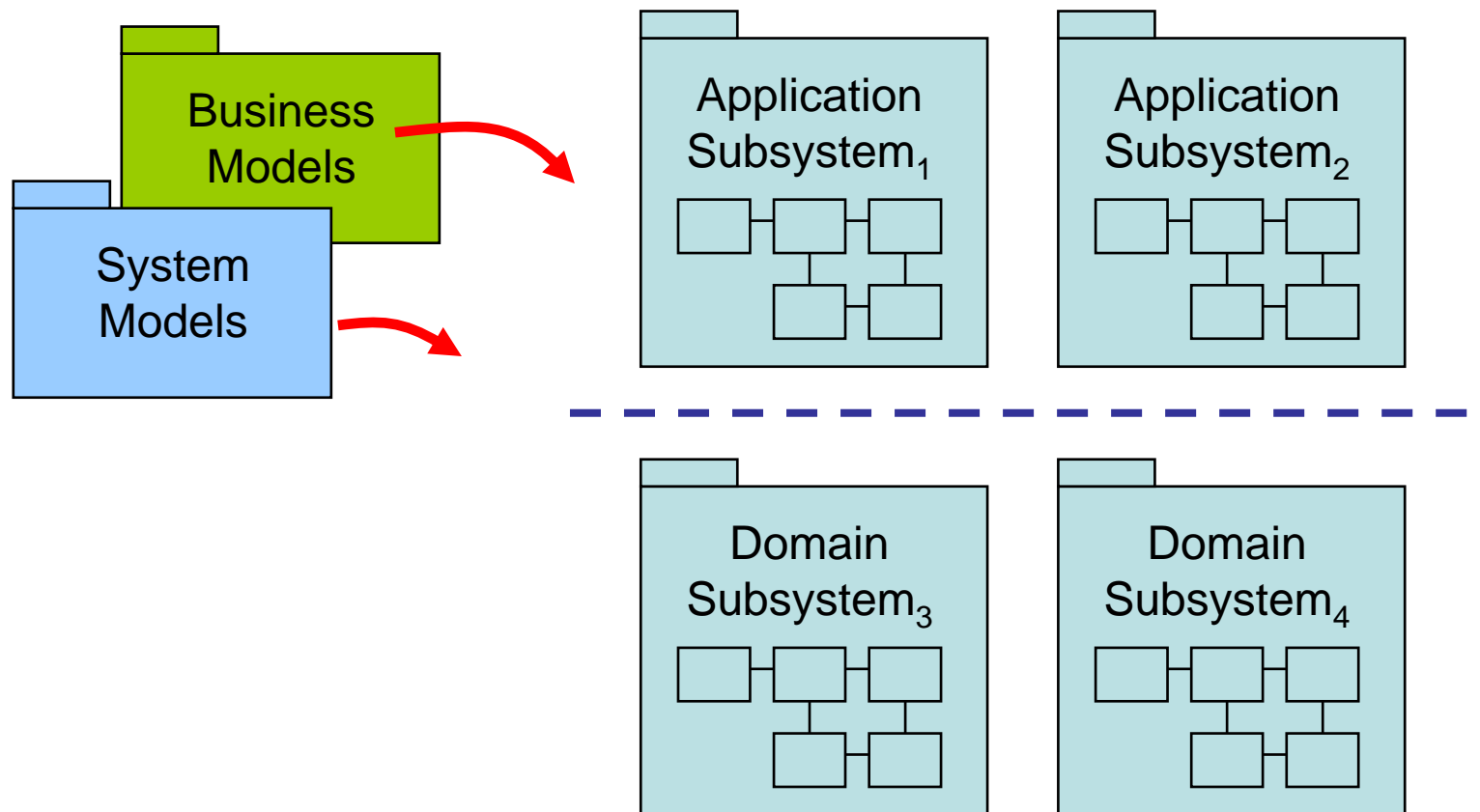
# What is “architecture” for?

- Architecture defines the road map
  - Describes the desired situation (business processes, IT systems) at some future time
  - When achieved should ensure seamless alignment of business and IT (a closed gap)
- Modeling is key
  - You need both business models and system models
  - You need clear relationships between business processes and systems
  - Modeling allows us to add further levels of detail as we proceed



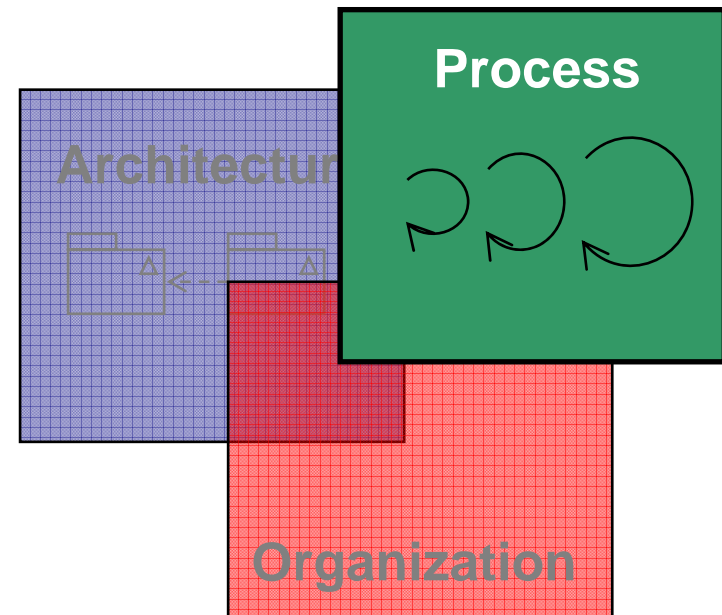
## Example – A sample architecture description

- The diagram shows the desired structure in layers
- These systems are identified through both business and system modeling



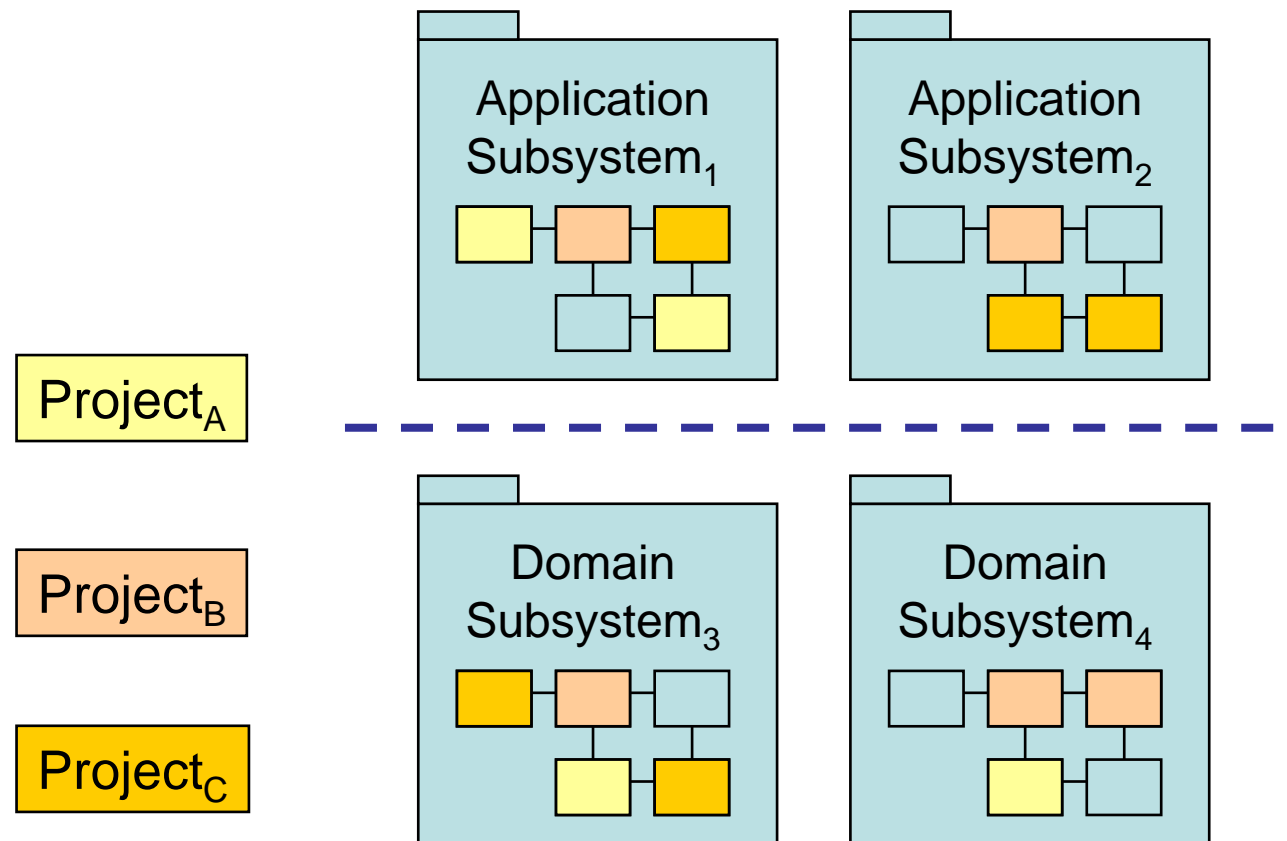
# What is “process” for?

- Process defines how the road map will be filled in
  - How you iteratively fill the desired structure project by project
  - Identify projects & dependencies, master schedule
- Iterative approach is key
  - You do not need to define every part of the desired structure in detail upfront – You add detail as you go
  - Roadmap is not just *paperware* but is refined based on actual experiences



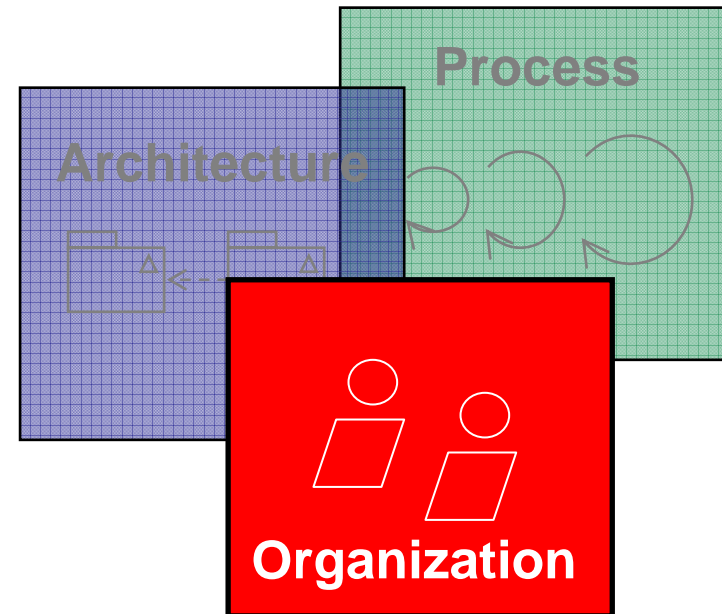
## Example – filling the desired structure project by project

- Each project will add or modify one or more IT systems
- Projects successively “fill in” applications and components
- After several projects we have a partially filled road map



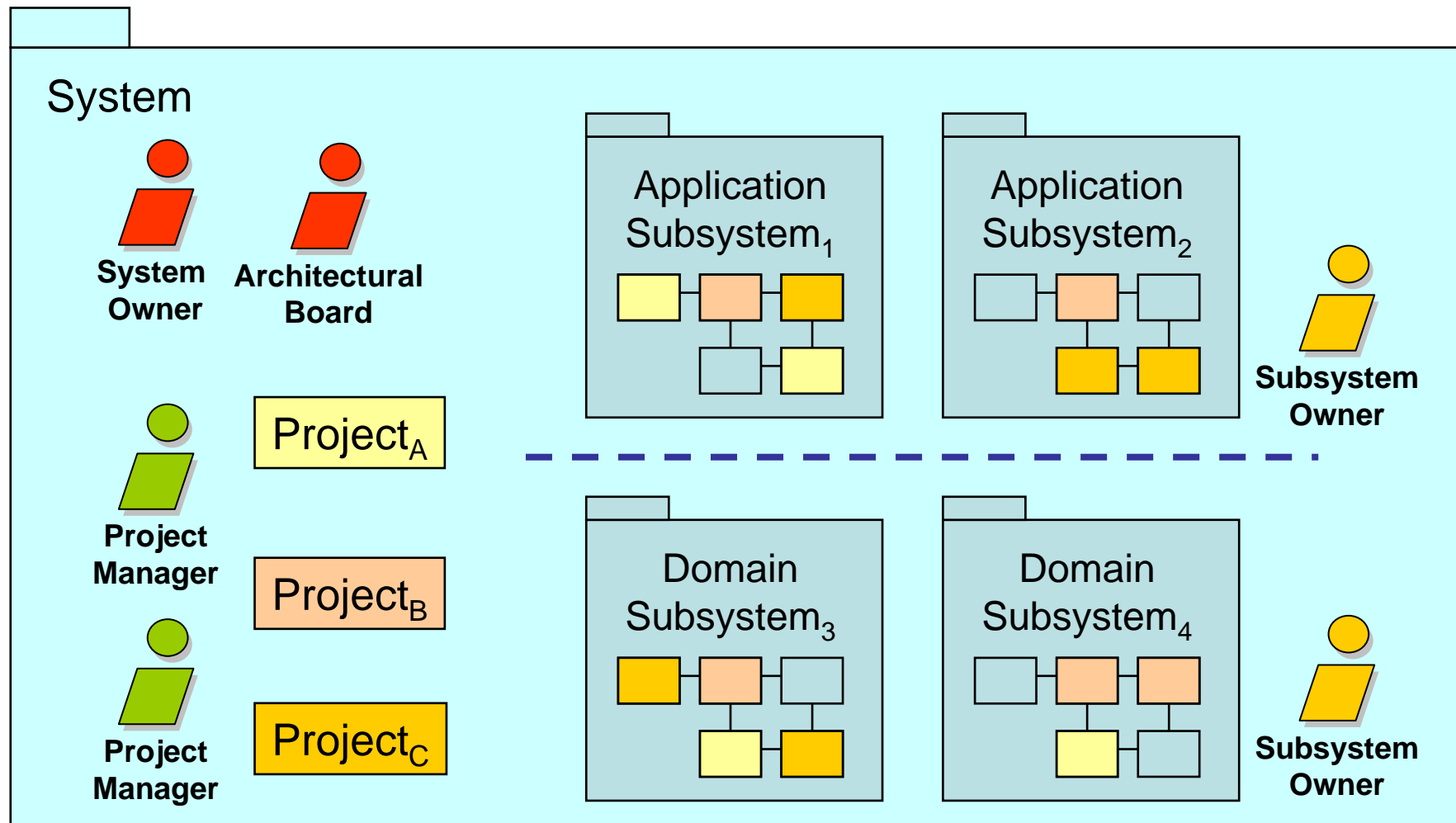
# What is “organization” for?

- Organization defines roles and responsibilities
  - Skills and teams needed, as well as staffing the IT organization
  - Should closely reflect the structure of the Architecture
  - Ensures everybody can work effectively and progress can be rapid
- Organization is aligned to architecture and process
  - Combination of project owners, component owners and architecture board



# Example – IT staff organized to achieve the desired structure

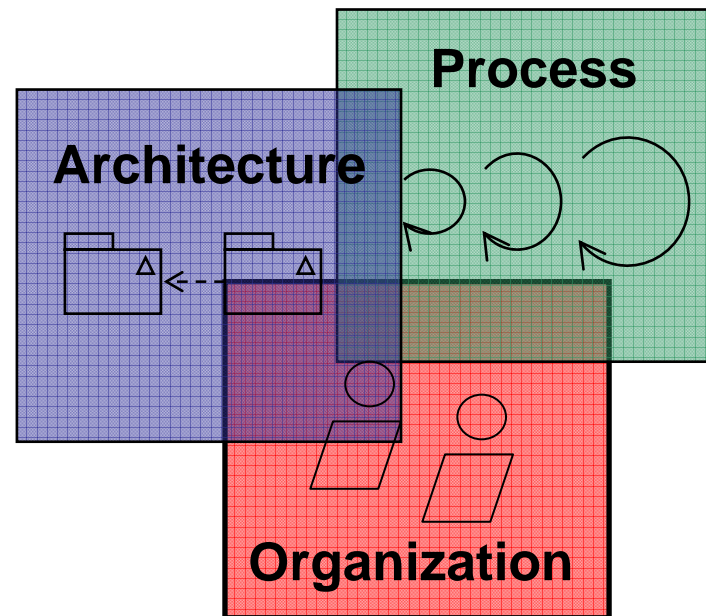
- Each project "bought" resources from subsystem owners
- No need for librarians or reuse agents



# Putting architecture, process and organization together

Architecture road map identifies building blocks

Projects gradually fills the building blocks



Organization aligned to architecture and process

# Agenda

- Gaps in enterprise IT
- Realities in enterprise IT
- Need for enterprise architecture
- Ingredients for a good enterprise architecture
- Our approach towards enterprise architecture
- Applying our approach in practice
- Turn enterprise architecture to reality

# Our approach to enterprise architecture

- Encompasses both business and system at different levels of abstraction

Business	System
External Perspective	External Perspective
Internal (abstract) Perspective	Internal (abstract) Perspective
Internal (concrete) Perspective	Internal (concrete) Perspective

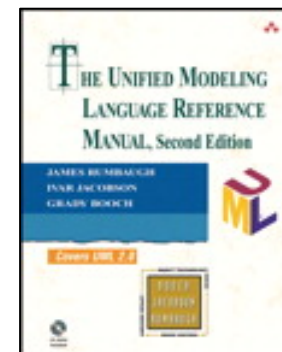
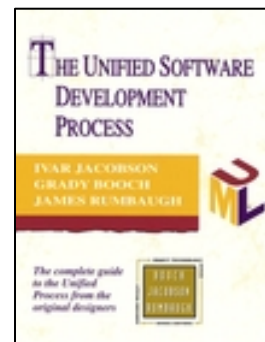
# Our approach to enterprise architecture

- Applies a use case driven approach
- Use case driven development is mature technique

<b>Business</b>	<b>System</b>
Business Use Case Model	Use Case Model
Business Analysis Model	Analysis Model
Business Design Model	Design Model

# Our approach to business modeling

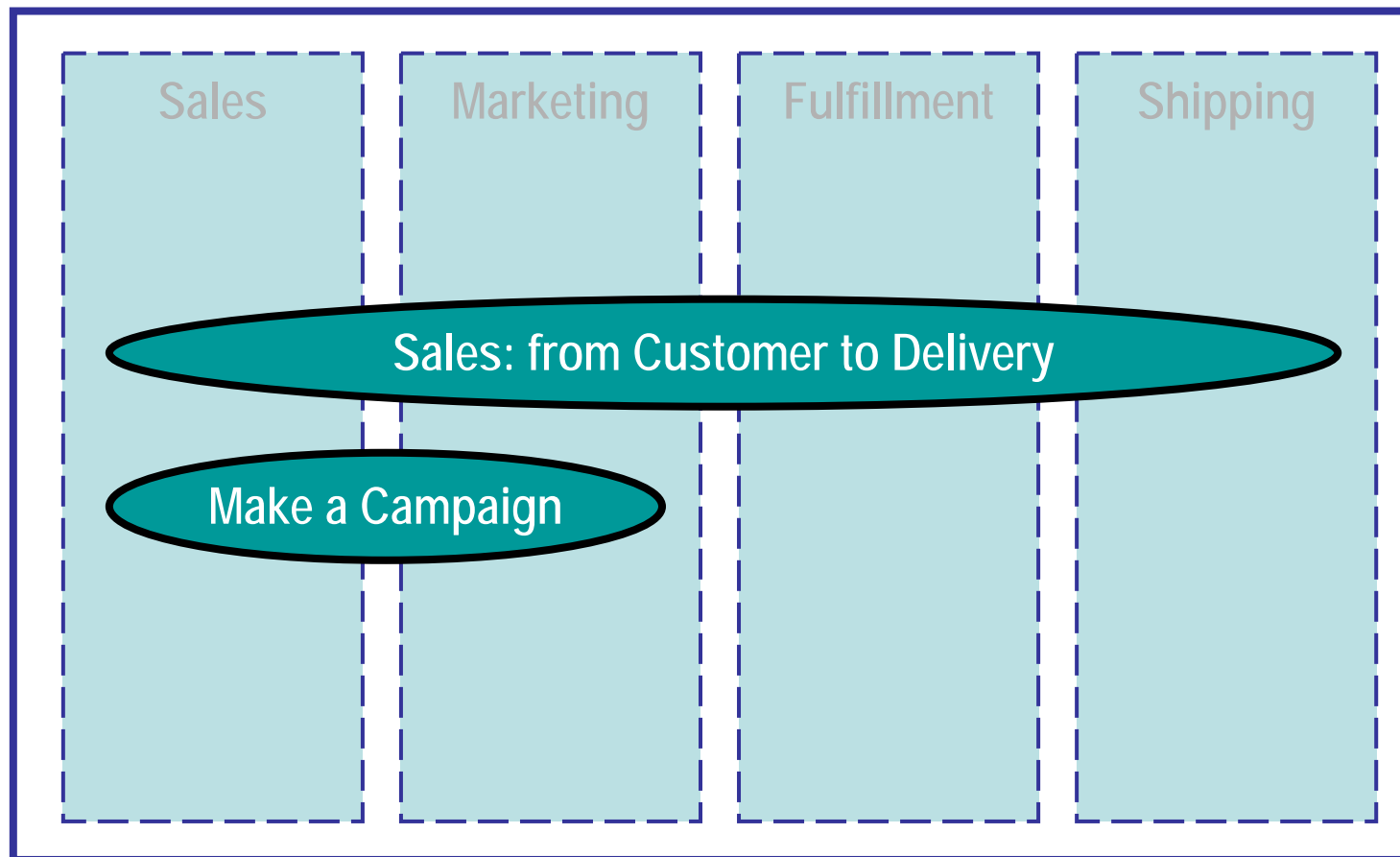
- Consistently and continually developed since 1994
- Based on “The Object Advantage”
- A key component of the Unified Process (UP)
- And of the Unified Modeling Language (UML)



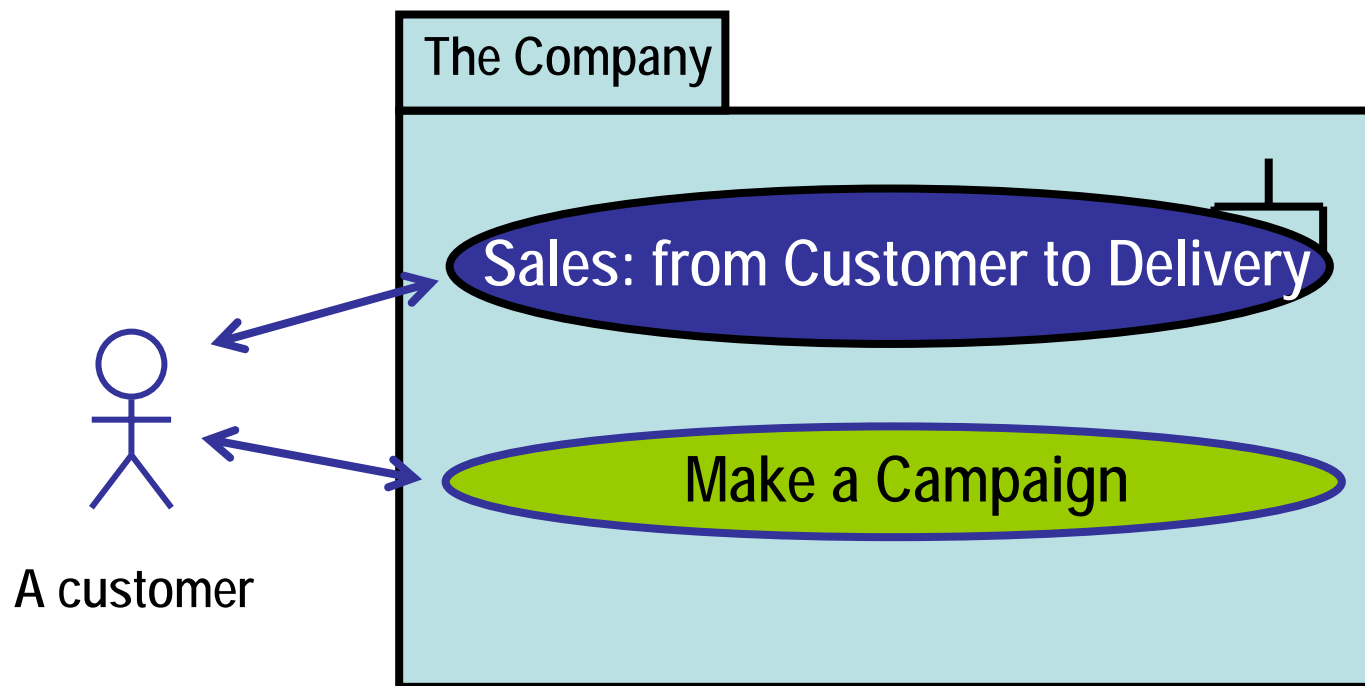
# The Example

- The Company offers business processes

## The Company

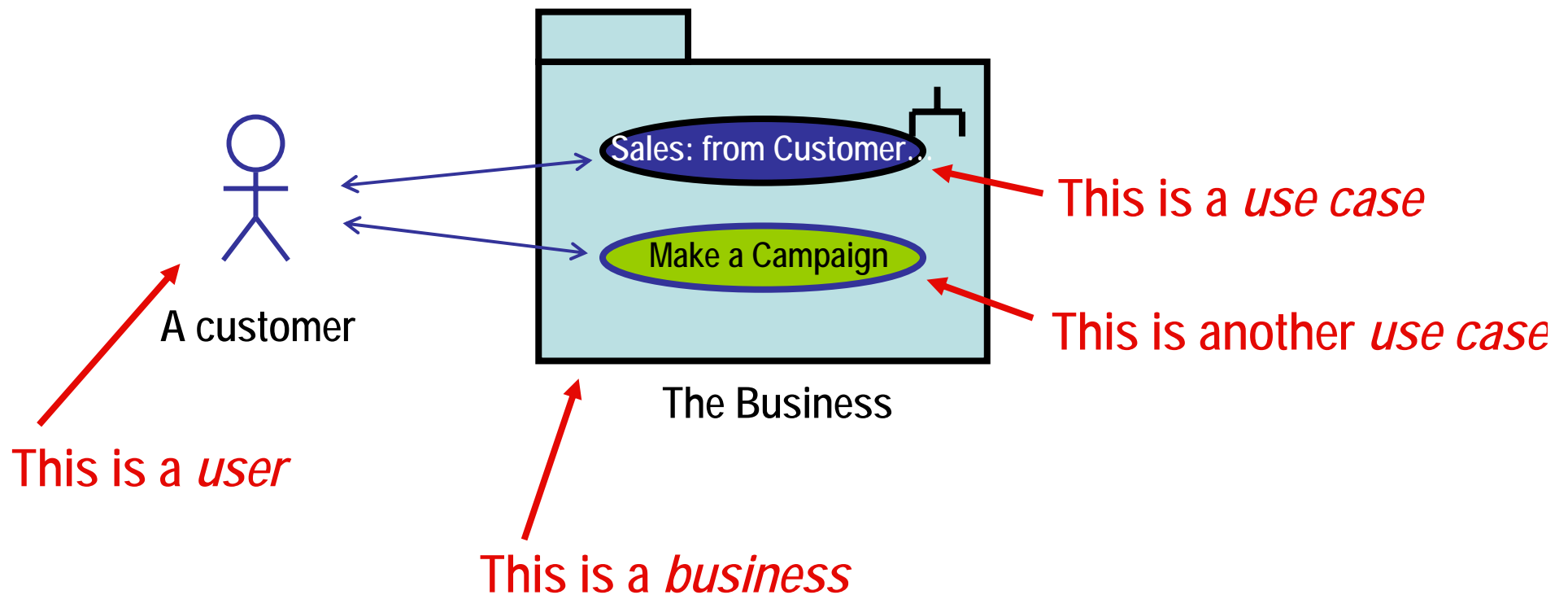


# The Example Using Business Modeling in RUP



## A Business Model – external view

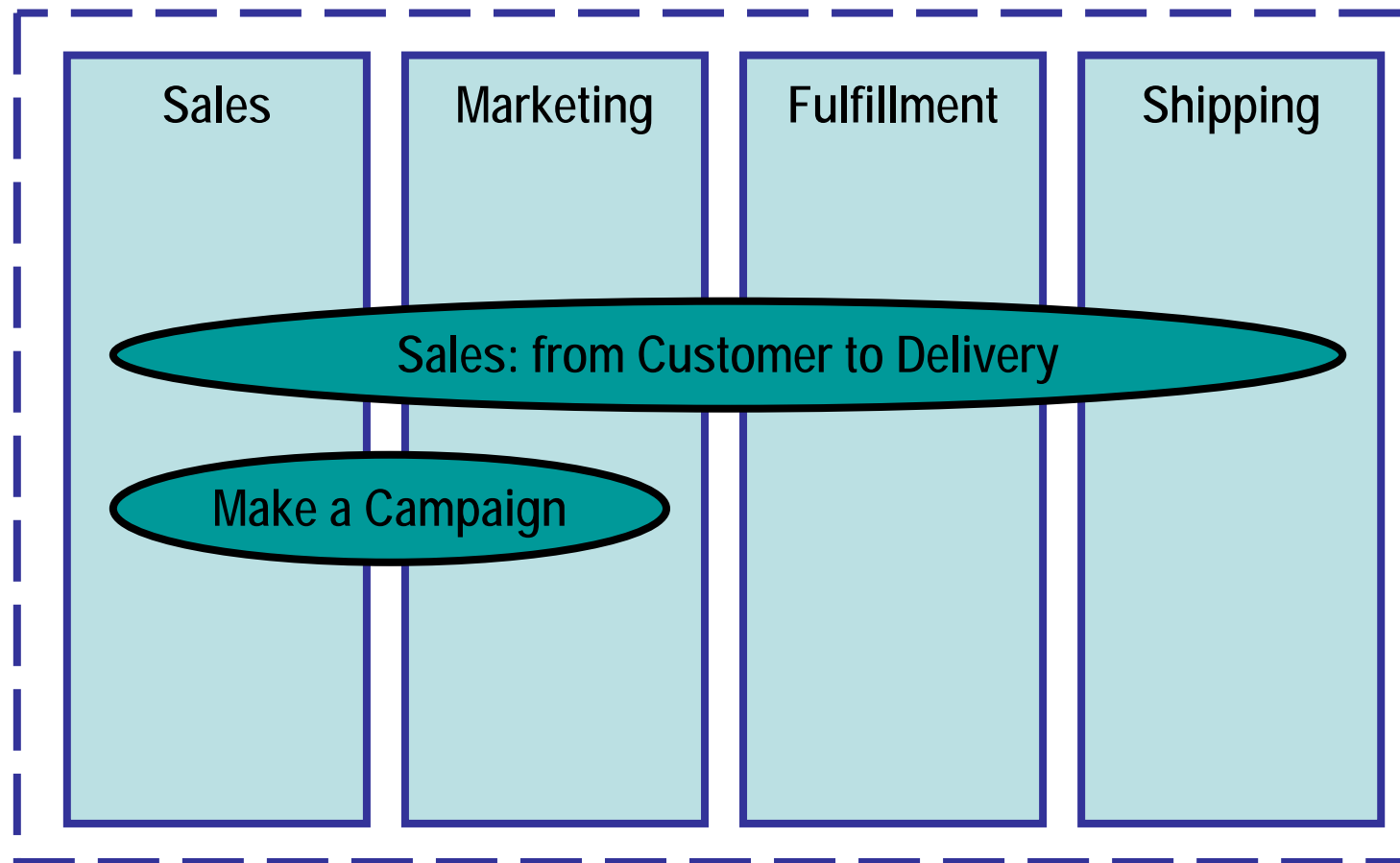
- The business is a system
- The business has users – customers or partners
- The business provides use cases = business processes
- This is an external view of the business



# The example taken inwards

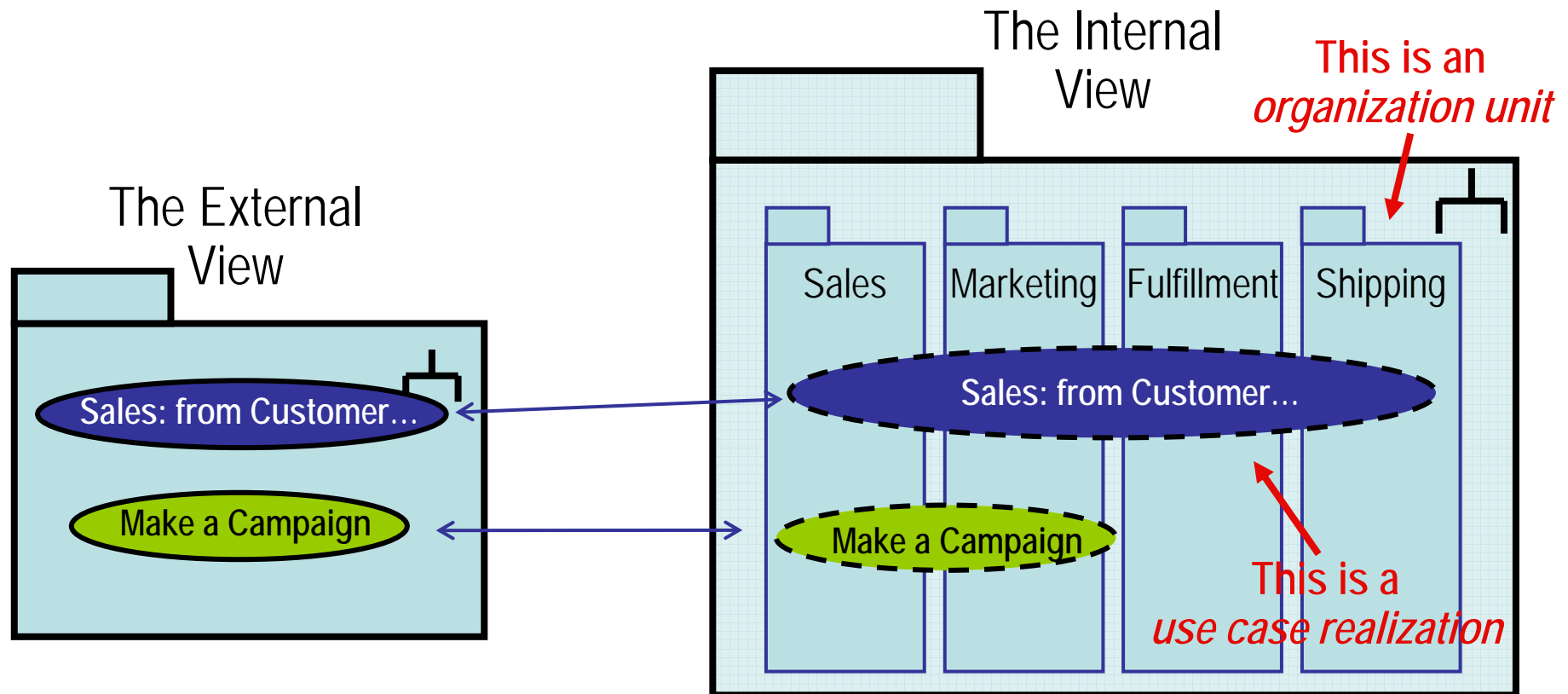
- Business Processes Cross Organization Units

## The Company



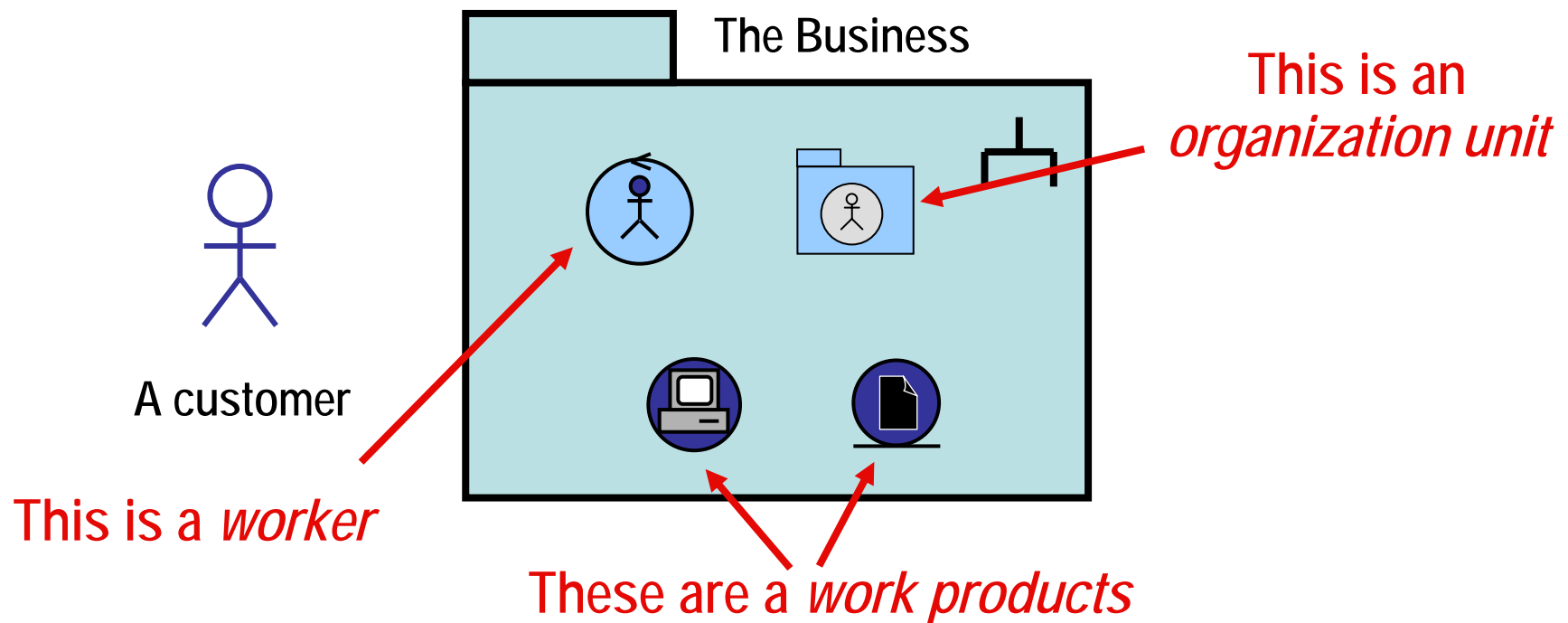
## Business model – internal view

- Use cases in the external view are realized as *use case realizations* in the internal view
- Use case realizations are collaborations among **organization units**



## Business model – internal view is more...

- The business contains (pick your own notation)
  - organization units,
  - workers,



## Merits of this approach to business modeling

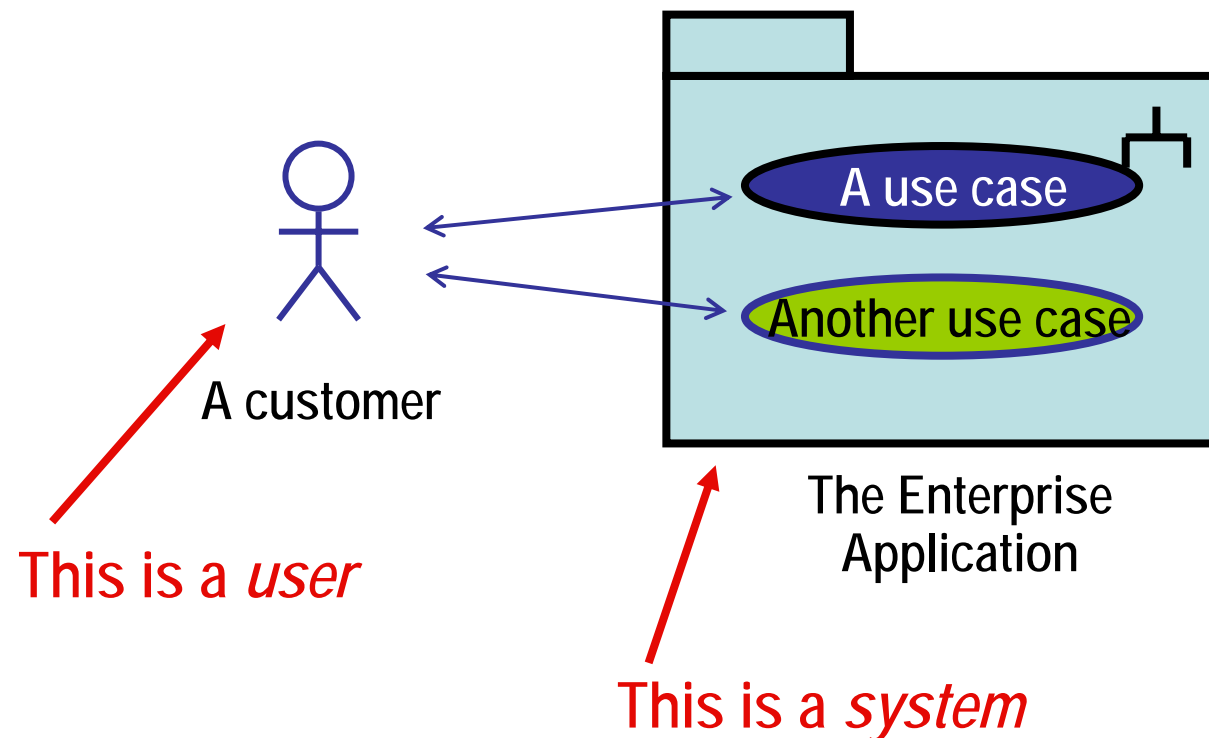
- It is based on a standard – the UML
- Use cases (and their collaborations) replace traditional process decomposition
  - Increased understandable since activities will be anchored in real roles, not floating around in semantic emptiness.
- The mapping from business modeling to system analysis is seamless

## Our approach to system design

- Consistently and continually developed since more than 30 years
- Based on experience from 1,000's of projects and 10,000's of developers
- It is component-based
- It means significant large scale reuse
- It uses the Unified Modeling Language (UML)
- It is the Unified Process

## System Design – external view

- The Enterprise Application is a *system*
- The system has *users*
- The system provides *use cases* = system processes
- This is an external view of the system



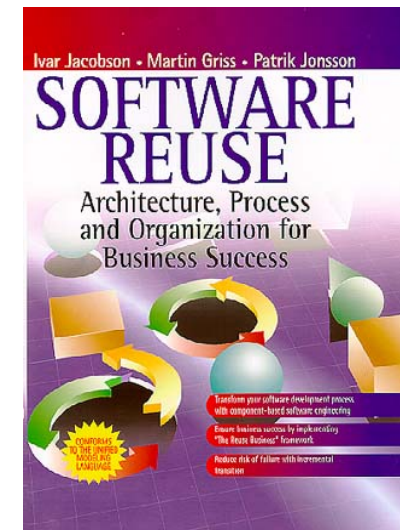
## System Analysis – internal view

It is similar to the internal view of Business Modeling

- A system contains *components* of different kinds – coarse-grained and fine-grained.
- The use cases of the system are realized as *collaborations* among these components.
- Helps in getting frictionless design and implementation

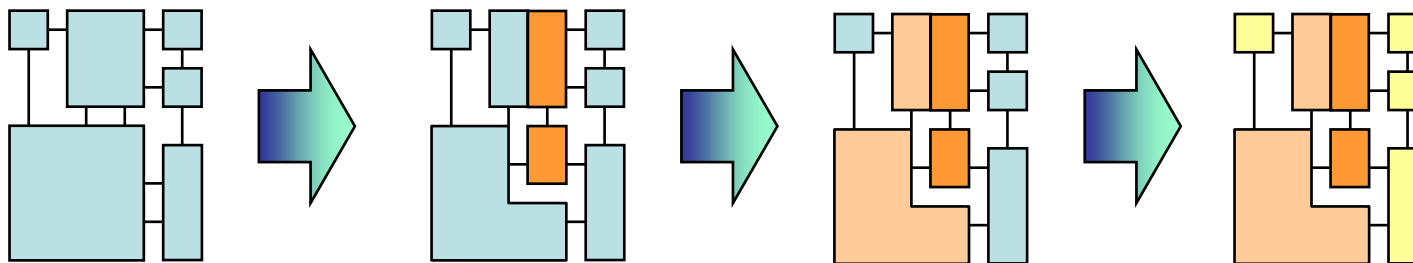
# Our Approach to Reuse

- Building a System-Of-Systems
- A proposed standard for reuse of assets
  - Reusable Asset Specification (RAS)
- Components in many shapes
  - Legacy systems
  - Package solutions
  - Web services



## Filling the details of the enterprise architecture

- The use case model and the analysis model is a road map
- The road map need not be fully detailed initially, as it will evolve over time
- The detailed design will be filled in over a period of time project by project



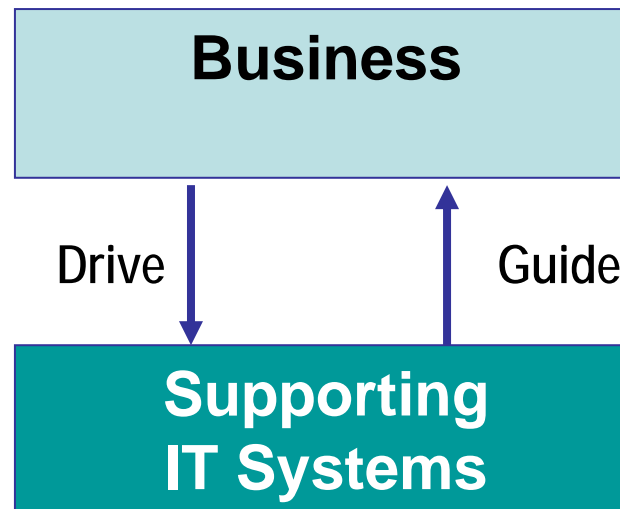
Remember: Enterprise architecture is not just about models, but also process and organization

# Agenda

- Gaps in enterprise IT
- Realities in enterprise IT
- Need for enterprise architecture
- Ingredients for a good enterprise architecture
- Our approach towards enterprise architecture
- Applying our approach in practice
- Turn enterprise architecture to reality

## From business to IT support system

- The Business must drive the IT support systems
- The available IT systems must guide the business
  - The business may have to be marginally adjusted to be supported by efficient IT systems
- The end-result is a balance between what is desirable (ideal) and what is available (real)



### The **Top-Down** Track

- "Think" about what you want your business to become
  - Business Modeling
- "Think" about how these business requirements drive your IT support system
  - It means Business Design

### The **Bottom-Up** Track

- Discover which applications are available
  - Get a System Design

### The **Balancing** Track

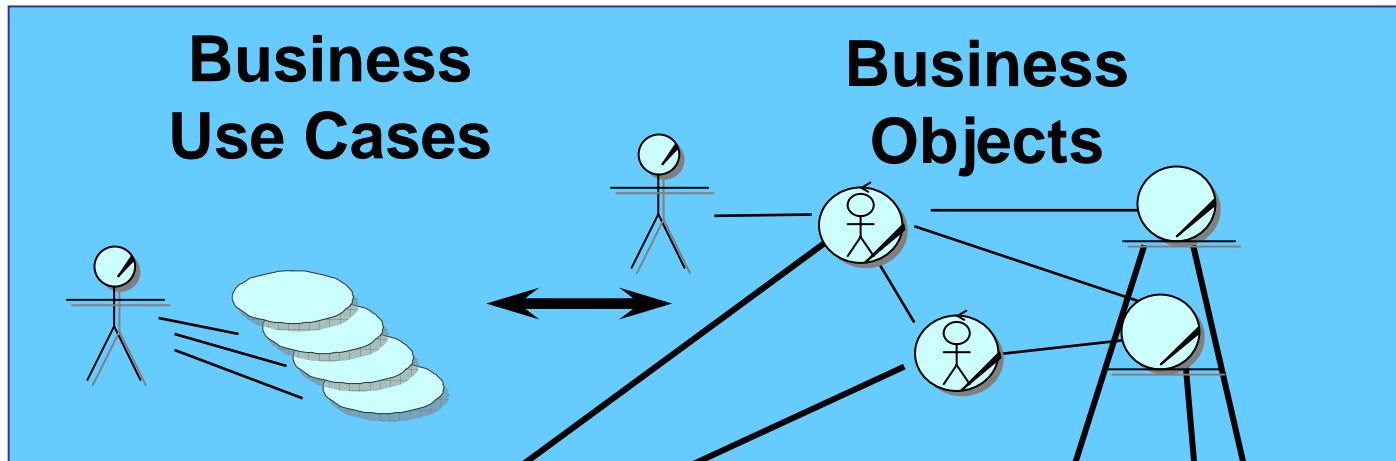
- ◆ Match what is desirable to what is available
  - It means decide on what to buy and what to make
- ◆ Make it happen!

## Is it worth doing all this?

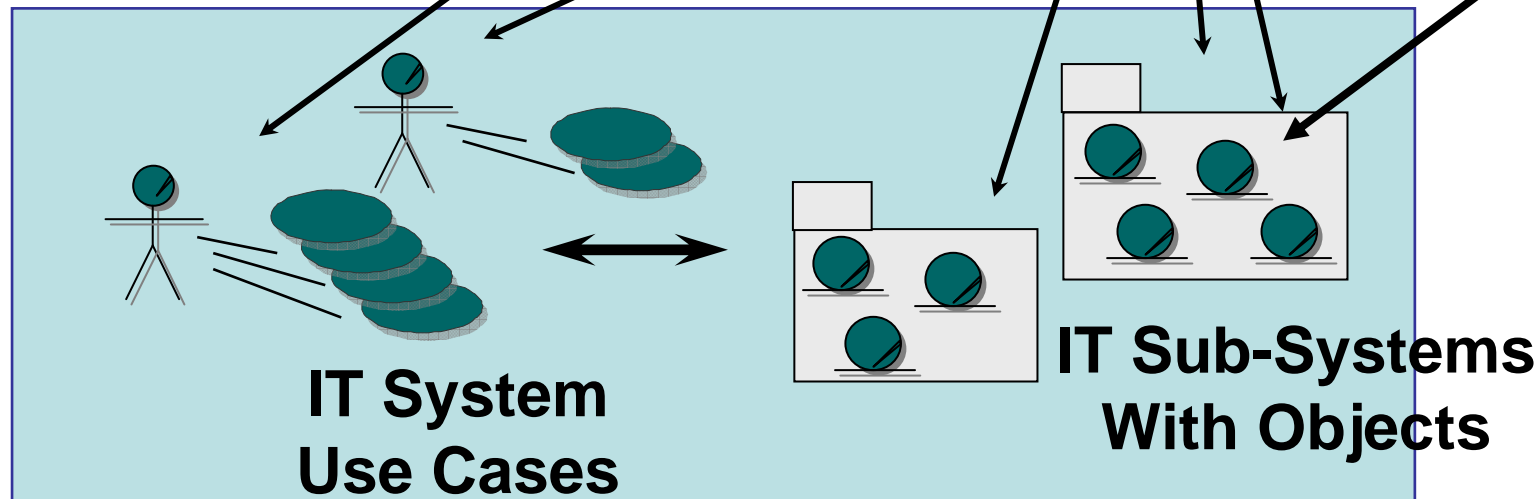
- Yes!
- The extent of modeling may vary from light to rigorous
  - It must always be done to adapt to change
- The three tracks – the Top-Down, the Bottom-Up and the Balancing – all happen within one or two iterations
- The success of your effort is based on understanding **what you want to get** and **what you can get** and on balancing these two perspectives
- It means modeling!

# The Top-Down Track: The Business Drives the IT System

## Business Model



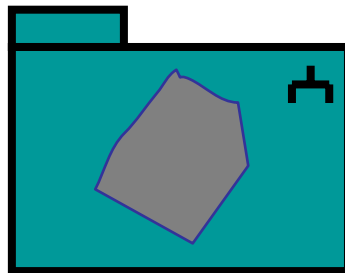
These are Applications



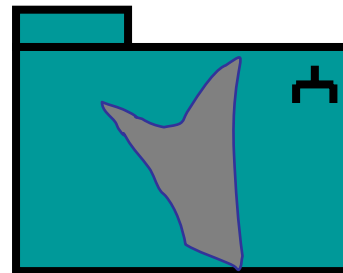
## Supporting IT System Models

## The Bottom-Up Track: available IT support guides the business

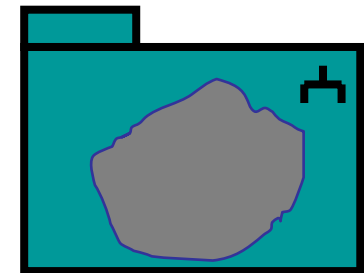
- First discover what You have?
  - Identify candidate software packages
- Put a wrapper around each one of them
  - A wrapper includes UML based descriptions of the package



Enterprise  
Resource  
Planning



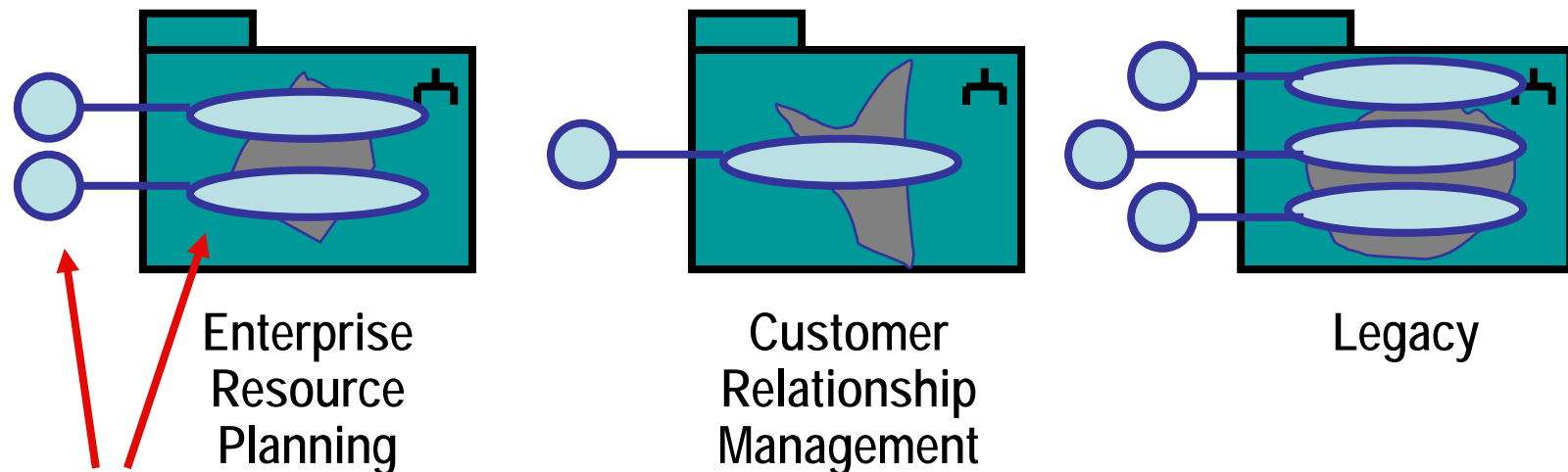
Customer  
Relationship  
Management



Legacy

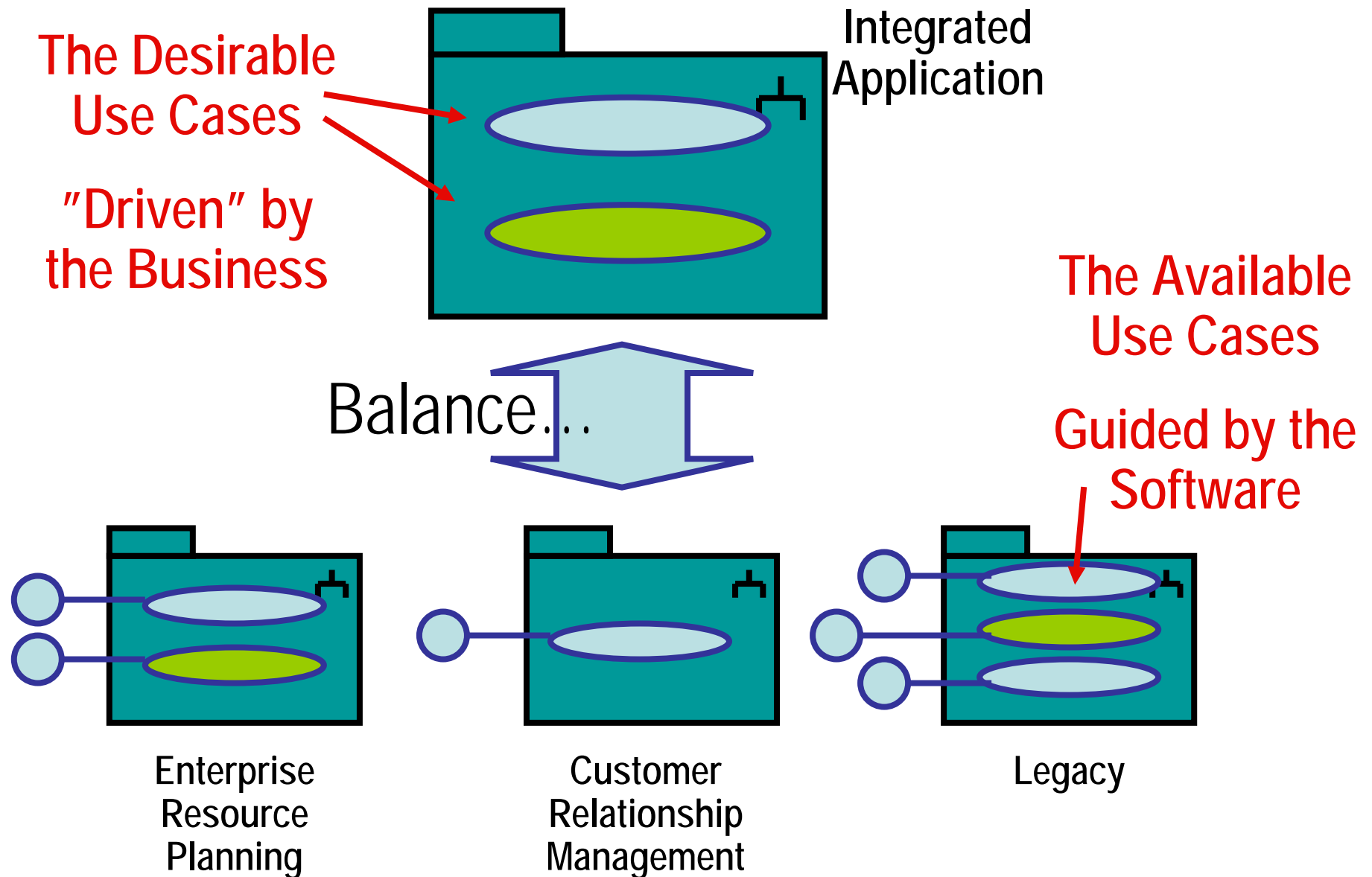
## The Bottom-Up Track: available IT support guides the business

- First discover what You have?
  - Identify candidate software packages
- Put a wrapper around each one of them
  - A wrapper includes UML based descriptions of the package
  - Most important it includes the use cases and the interfaces



Available Use Cases and Interfaces

# The Balancing Track 1

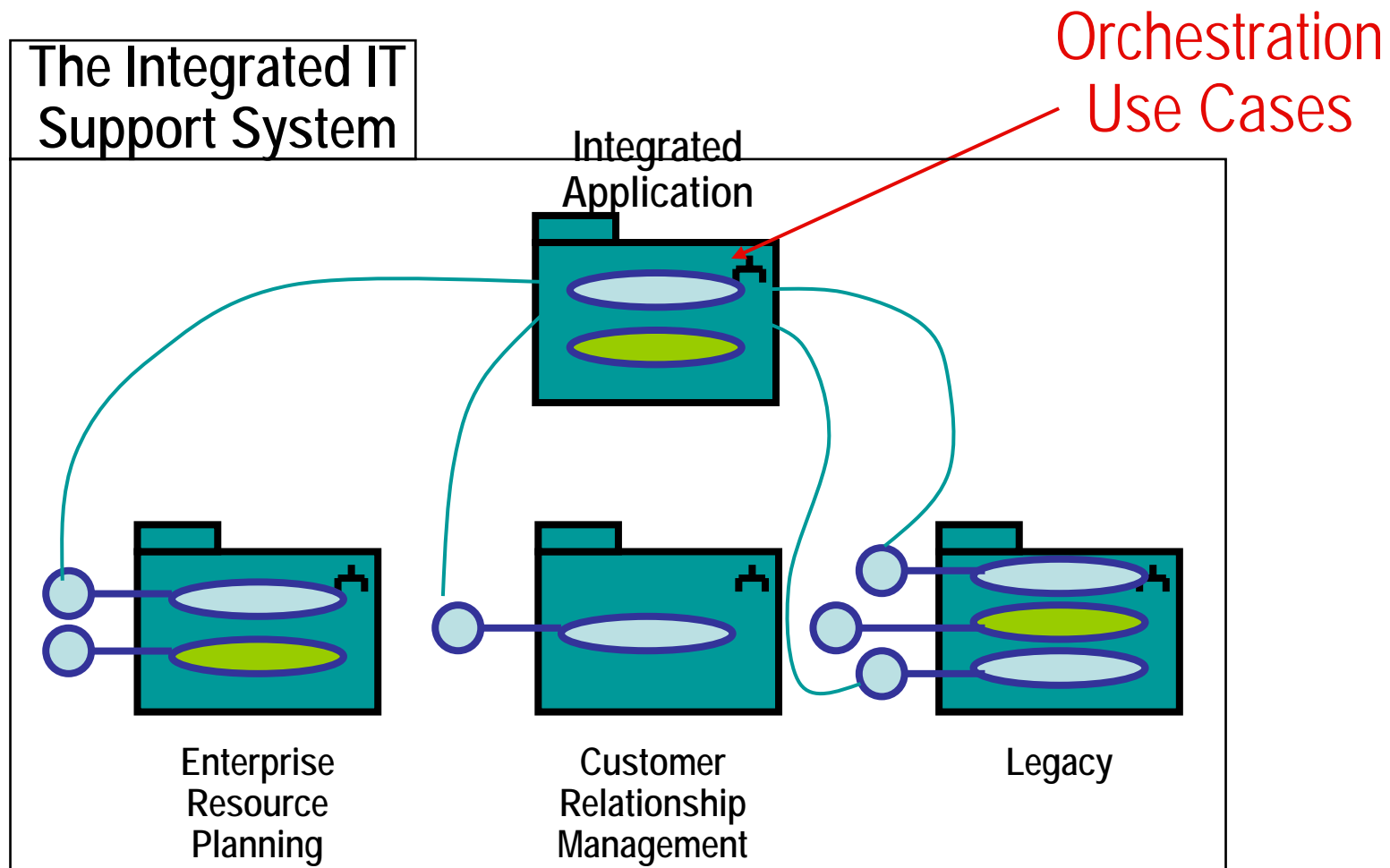


## The Balancing Track 2

- Compromise what is desirable with what is available (or you have to build the software yourself!)
  - It is a systematic way to find what to buy and what to make
- Once an efficient balance (or match) is achieved,
  - Change your desired business model to one that can be supported by your new IT system, and
  - Change your available software by
    - Changing your legacy system,
    - Buy and extend available applications, and in some cases – if necessary
    - Make your own new applications.

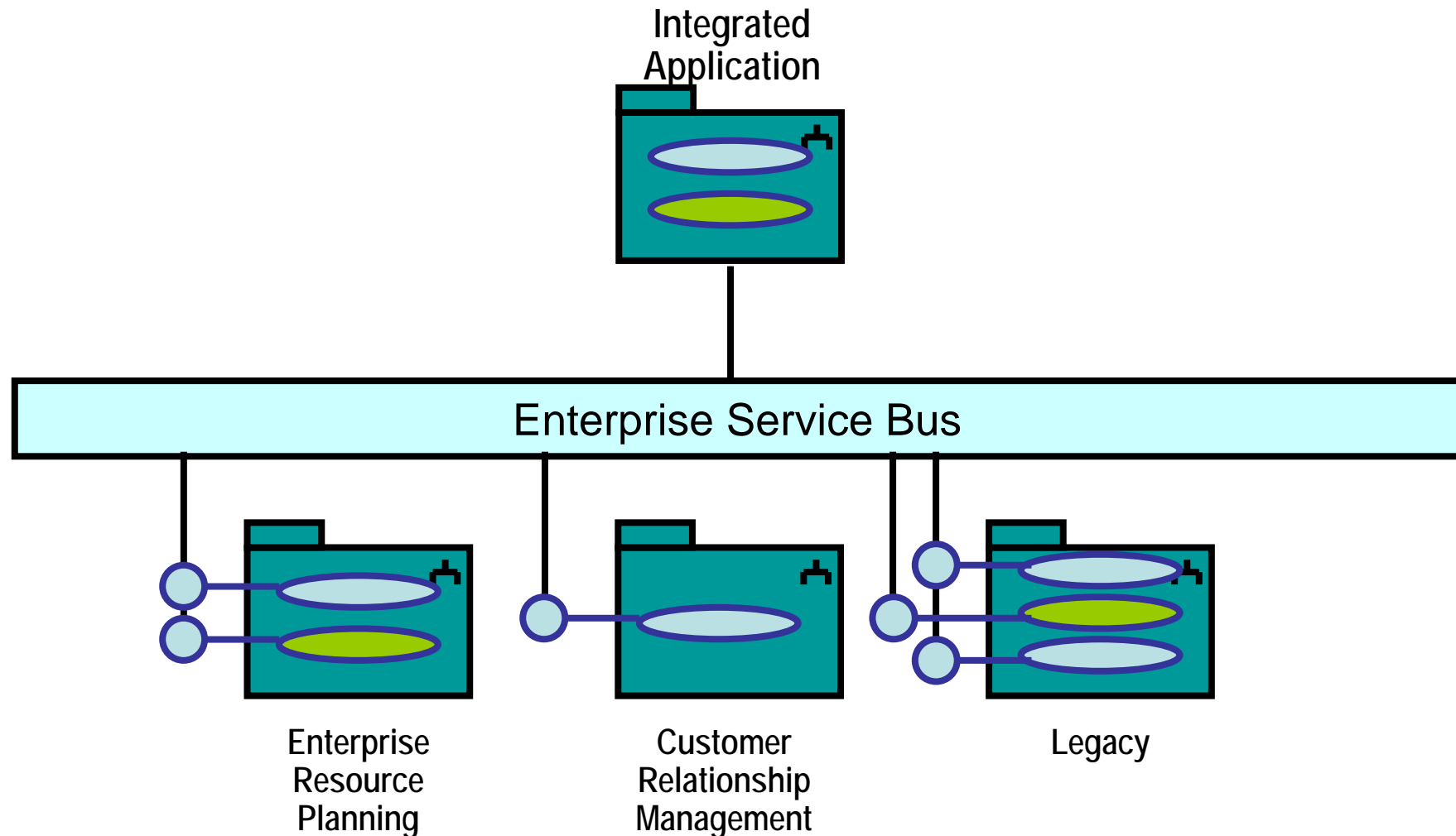
## The Balancing Track 3: resulting integrated IT support system

- Let the hub orchestrate the use cases of the application to achieve the use cases of the Integrated IT support system



## The Balancing Track 3: result on an Enterprise Service Bus

- Use available connection technology, e.g. enterprise service bus

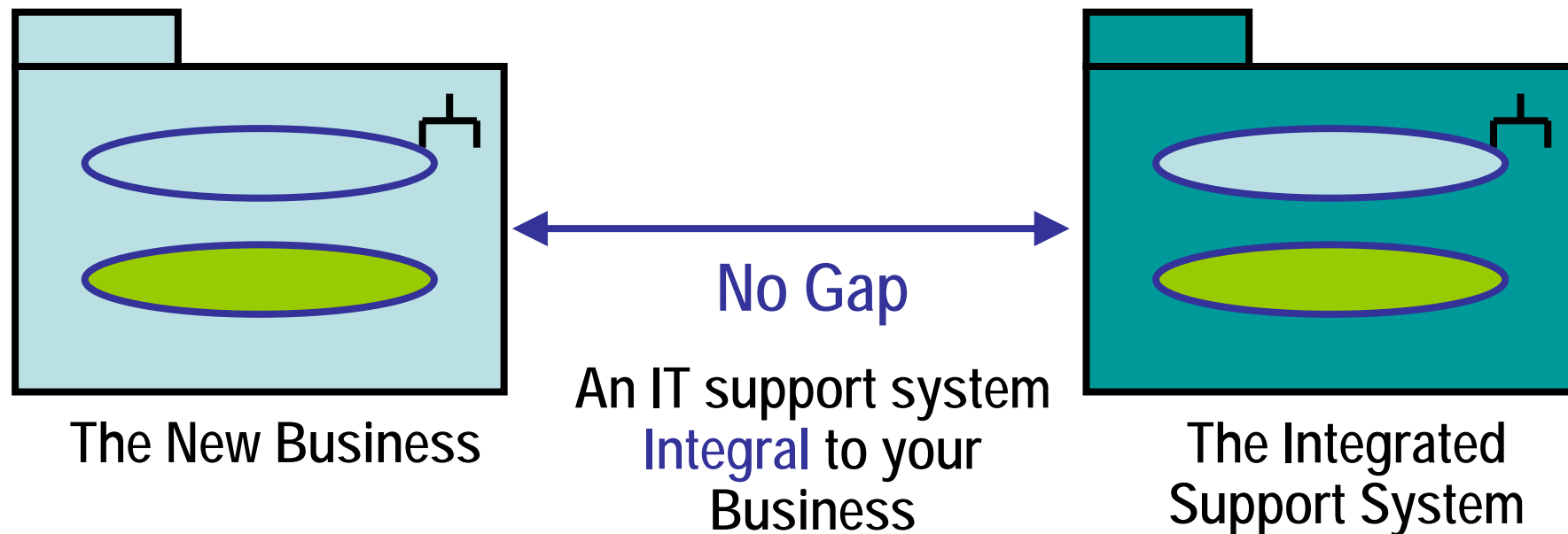


# Agenda

- Gaps in enterprise IT
- Realities in enterprise IT
- Need for enterprise architecture
- Ingredients for a good enterprise architecture
- Our approach towards enterprise architecture
- Applying our approach in practice
- Turn enterprise architecture to reality

# Closing The Gap

- We started from what we think we wanted
- We discovered what we knew was available
- We balanced what we want with what we can get
- We made it happen!



## Modern IT means

- Driving your enterprise architecture from the needs of your Business
- It means understanding your business and your available applications
  - Understanding means modeling. It means UML! 😊
- Iteratively realize the models project by project
- With IT organization aligned to architecture and process
- It means substantial savings for your company
  - No huge upfront investments
  - Substantial REUSE across your company
- It means a common IT-language across your company

You Close The Gaps!

# References

- The UML Books (Booch, Jacobson, Rumbaugh with Addison Wesley)
  - The UML User Guide
  - The UML Reference Manual
  - The Unified Software Development Process
- The Rational Unified Process
  - <http://www.rational.com/products/rup/index.jsp>
  - The Rational Unified Process - An Introduction Philippe Kruchten (Addison Wesley)
- Reusable Asset Specification
  - [www.rational.com/eda/index/jsp](http://www.rational.com/eda/index/jsp)

## References cont'd

- On Use Cases
  - Object-Oriented Software Development--A Use Case Driven Approach (Addison Wesley)  
Jacobson et al, Addison Wesley Longman (1992)
- On Business Modeling
  - The Object Advantage: Business Process Reengineering with Objects (Addison Wesley)  
Jacobson et al, Addison Wesley Longman (1994)
- On Systems of Systems
  - Software Reuse: Architecture, Process and Organization for Business Success (Addison Wesley)  
Ivar Jacobson, Martin Griss & Patrik Jonsson, Addison Wesley Longman (1997)