

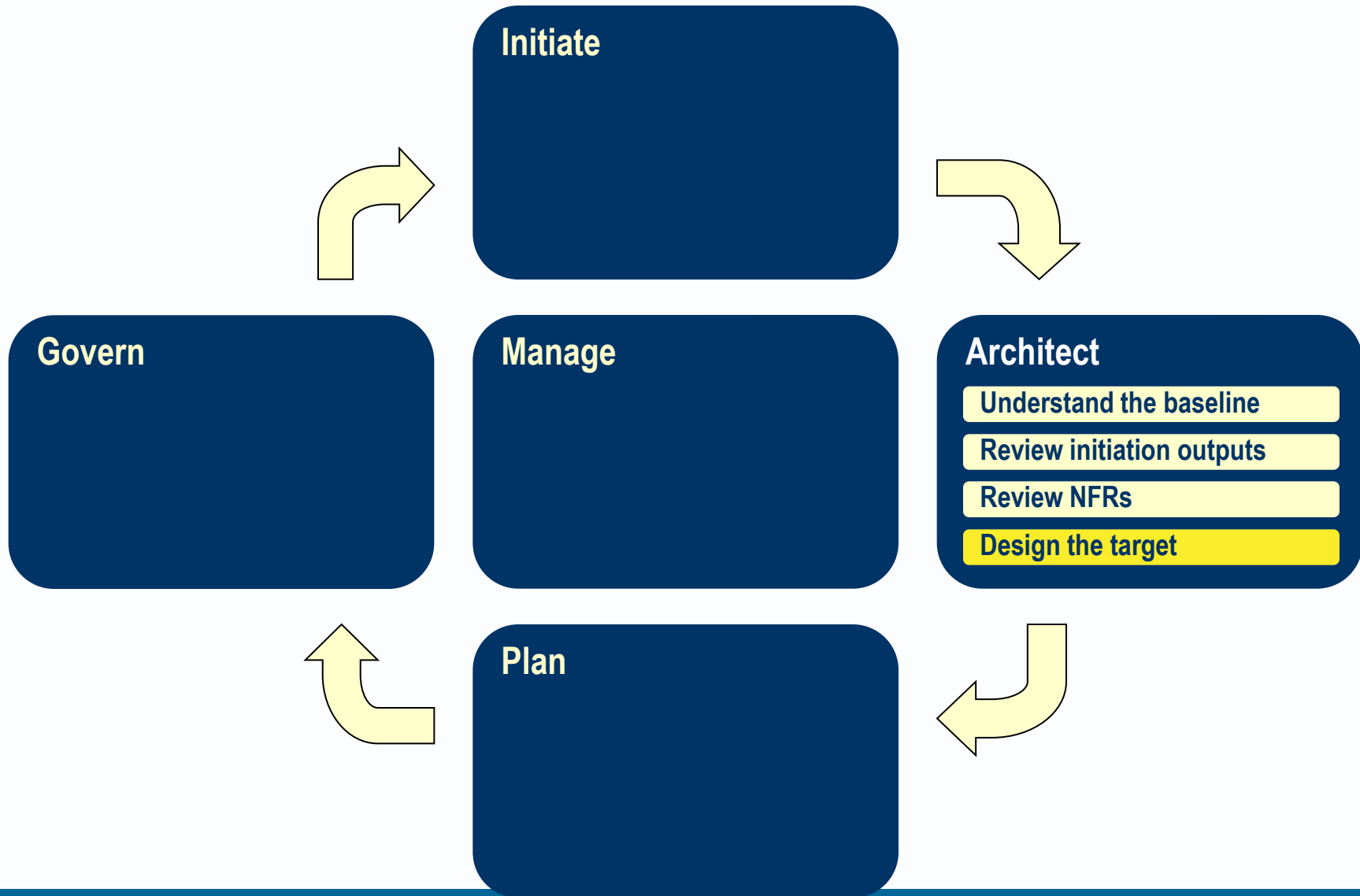
# Avancier Methods (AM)

## Solution level Business Architecture

Scenario driven analysis and design

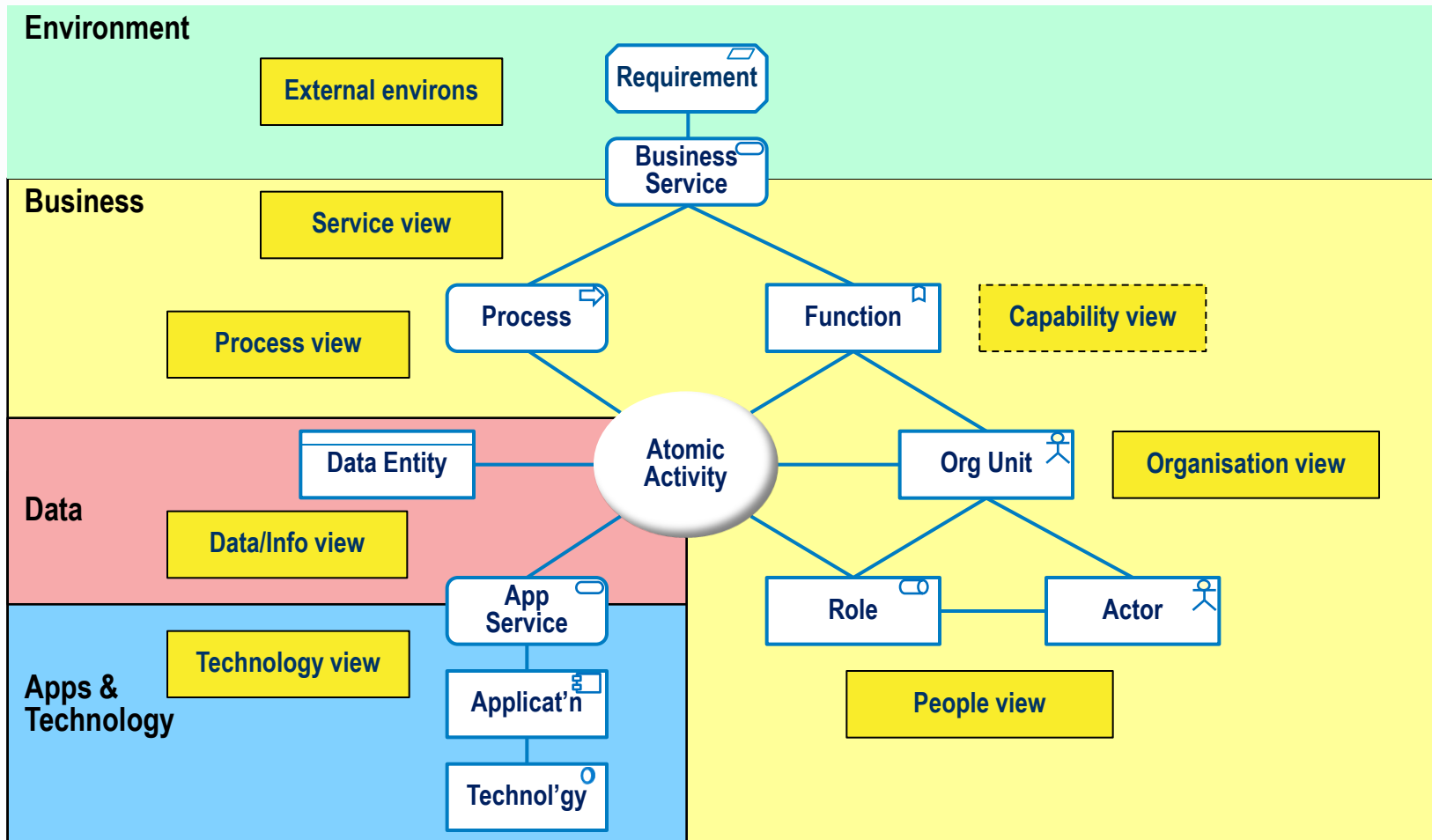
It is illegal to copy, share or show this document  
without the written permission of the copyright holder

# Where in the AM process?



# Skills Framework for the Information Age suggests 7 views

SFIA defines EA and BA as inter-relationships between 7 views (Capability view added below)

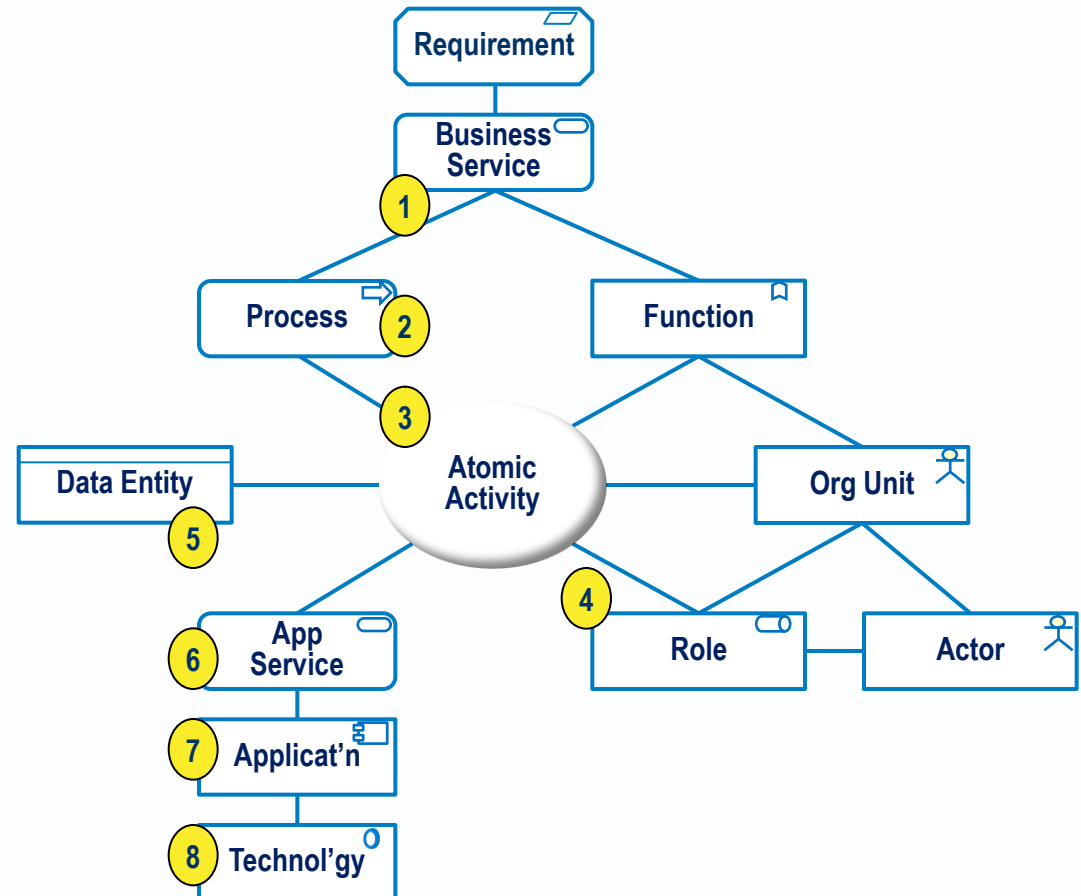


# Scenario-driven analysis and design

1. Identify required Business Service
2. Encapsulate Process as a Service
3. Decompose to OPOPOT activities
4. Assign activities to Roles
5. Define Data created and used
6. Define App Services needed
7. Define Applications
8. Define Infrastructure

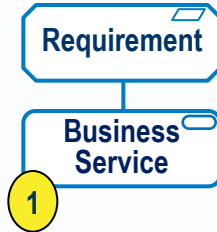
## Note:

- ▶ Business before technology
- ▶ External before internal
- ▶ Behaviour before structure
- ▶ “The level and rigor of decomposition needed varies”



# 1. Identify required Business Service

- ▶ EA as Strategy: “Define your operating model”
  - Identify processes that distinguish you competitively
  - Envision your customer’s experience as it ought to be
  - Decide how you your company will grow
  - Define services to be provided by the process or system to be designed
  
- ▶ E.g. The required business service is
  - **The capture of an order by a salesman at a customer’s premises**



## 2. Encapsulate Process as a Service

- ▶ Define end and/or output of the process - and post conditions
- ▶ Define trigger and/or input to the process - and preconditions



Order capture		
Signature	Name	Order Capture
	Input / trigger	Salesman given sales visit details
	Output / result	Salesman confirms order to interested parties
Functional rules	Preconditions	Sales visit to customer premises has been agreed and scheduled
	Post conditions	Order and Customer Details captured
Non-Functional requirements	Duration	2 days
	Throughput	10 per minute
	Commission	30 percent of order value

### 3. Decompose to OPOPOT activities

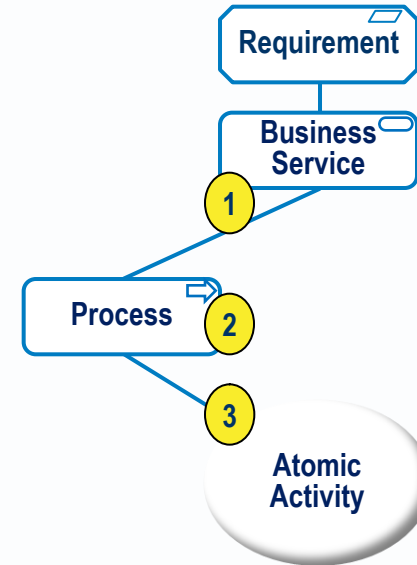
- ▶ Initially, define the straight-thru or happy path to the desired result

**Precondition: Sales visit to customer premises has been agreed and scheduled**

#### **Process**

- 1 Initiate sales process with the customer**
- 2 Discuss customer requirements**
- 3 Work with customer to create a product configuration**
- 4 Verify that the desired configuration can be delivered**
- 5 Determine price of requested configuration**
- 6 Confirm customer desire to purchase**
- 7 Place an order**
- 8 Capture customer signature**

**Post condition: Order captured**



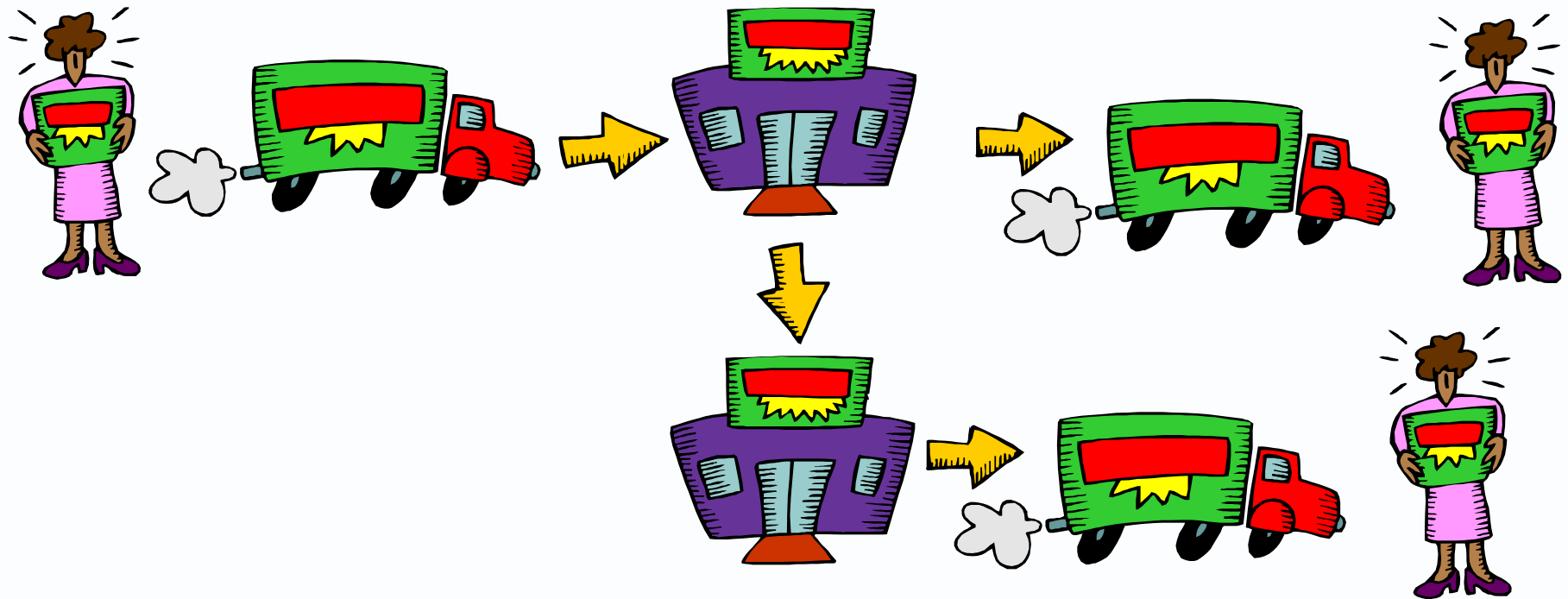
## Sooner or later, detailed analysis must be done

- ▶ Given the straight-thru or happy path
  
- ▶ Examine every step
  - What else could happen?
  - What else could an actor do - other than what is expected?
  - What if they don't do anything at all? Is there a time out?
  - What could wrong?
  - What resources created or used at this step are vulnerable to being exposed, lost or corrupted?
  - What security measures need to be put in place?
  
- ▶ These questions may be asked at any and every level of process decomposition, but here, the focus is on the first pass design.

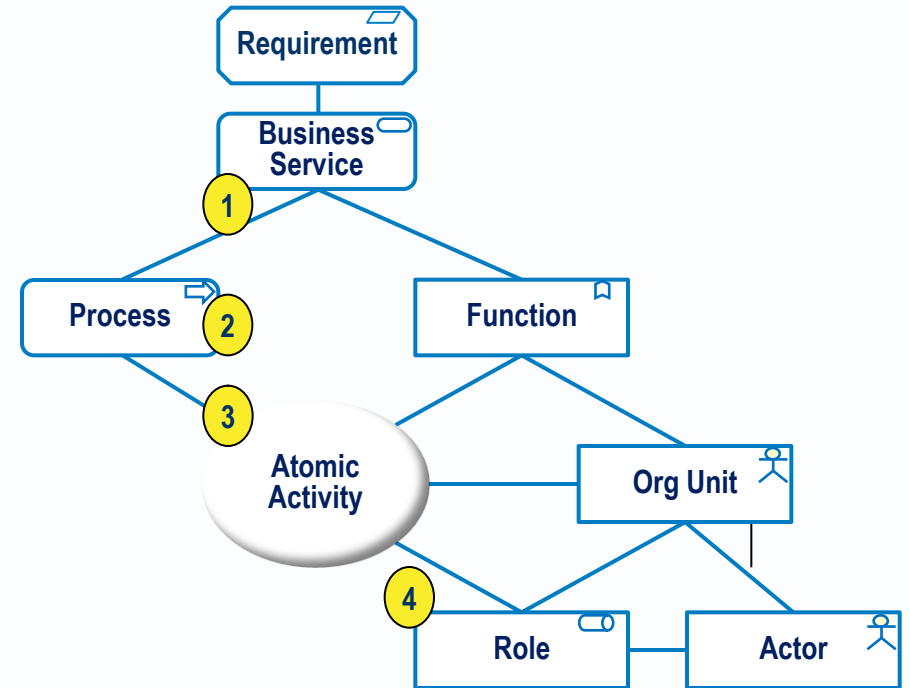


# More pictorial style

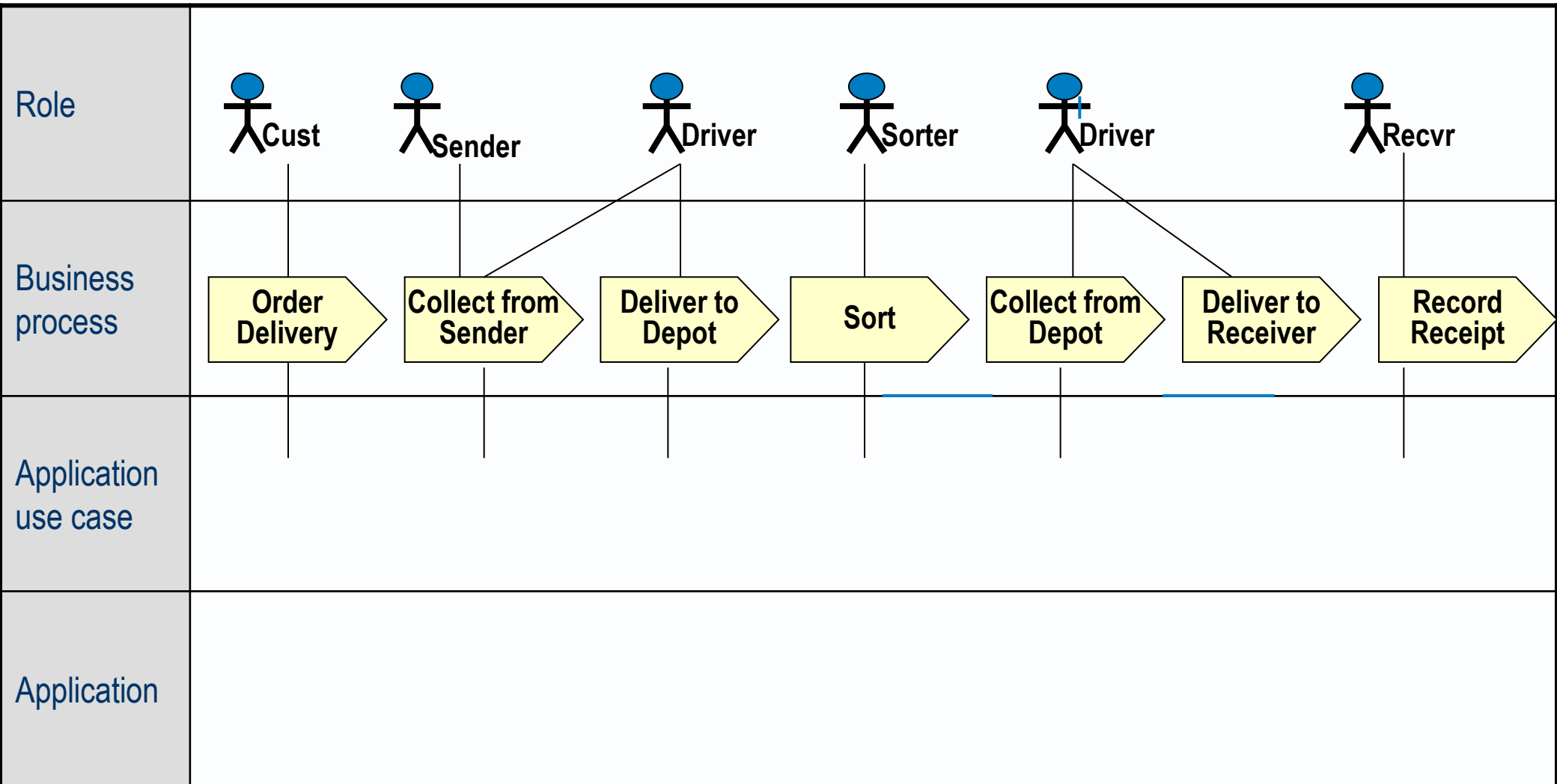
- ▶ You don't have to draw a formal process flow chart to begin with
- ▶ People like pictures



# 4. Assign activities to Roles



# Assign activities to Roles



# Assign activities to Roles

Precondition: Sales visit to customer premises has been agreed and scheduled	Human actors (roles)	
Process	Customer	Sales person
1 Initiate sales process with the customer	Accept sales person	Greet customer
2 Discuss customer requirements	Explain problems & requirements	Listen
3 Work with customer to create a product configuration	Select one option based on capabilities	Show product and configuration options
4 Verify that the desired configuration can be delivered		Show availability to customer
	Accept	Show delivery date to customer
5 Determine price of requested configuration	Accept	Accesses price system and Shows price to customer
6 Confirm customer desire to purchase	Accept	Show offer
7 Place an order		Captures order details and prints out fax
8 Capture customer signature	Sign	Requests signature
Post condition: Order captured		

## 5. Define Data created and used

Define data groups or items that each process step

- a) Needs or uses
- b) Creates or updates

Process	Data created and used
1 Initiate sales process with the customer	
2 Discuss customer requirements	
3 Work with customer to create a product configuration	Product description Configuration assembly
4 Verify that the desired configuration can be delivered	Configuration availability Delivery date
5 Determine price of requested configuration	Price of configuration
6 Confirm customer desire to purchase	
7 Place an order	Customer details
8 Capture customer signature	Paper copy of order Signature

Requirement

Business Service

1

Process

2

3

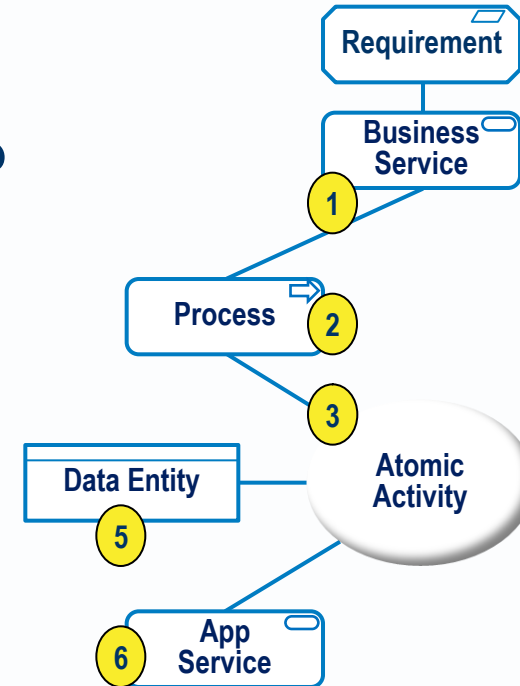
Data Entity

5

Atomic Activity

## 6. Define App Services needed

- ▶ Identify steps at which actors use applications and technologies to do the activity and create/use the data.
- ▶ Define the services needed from applications at each step
- ▶ List these as use cases



# Define Application Services needed



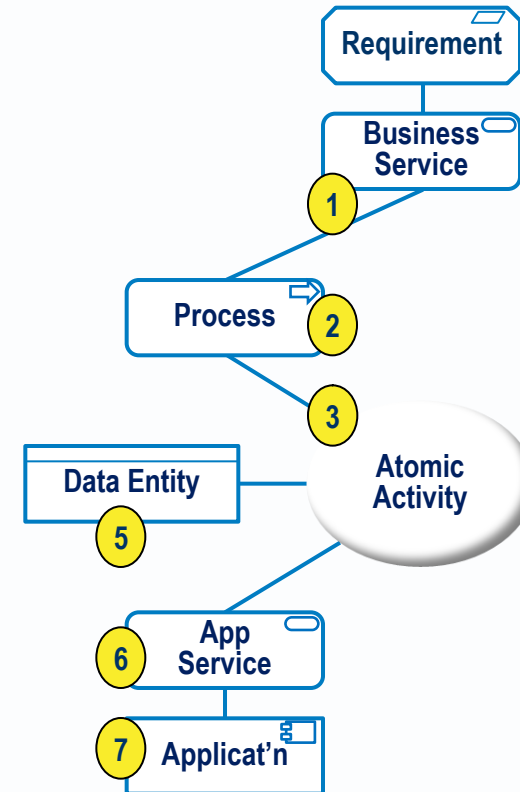
Avancier

Precondition: Sales visit to customer premises has been agreed and scheduled	Human actors (roles)		Computer actors (roles)
	Customer	Sales person	Client app use case
<b>Process</b>			
1 Initiate sales process with the customer	Accept sales person	Greet customer	
2 Discuss customer requirements	Explain problems & requirements	Listen	
3 Work with customer to create a product configuration	Select one option based on capabilities	Show product and configuration options	Get product description Assemble configuration
4 Verify that the desired configuration can be delivered		Show availability to customer	Check configuration availability
	Accept	Show delivery date to customer	Get delivery date
5 Determine price of requested configuration	Accept	Accesses price system and Shows price to customer	Price configuration
6 Confirm customer desire to purchase	Accept	Show offer	
7 Place an order		Captures order details and prints out fax	Enter order Get fax response
8 Capture customer signature	Sign	Requests signature	
Post condition: Order captured			

## 7. Define Applications

Given the required use cases...

- ▶ Identify applications to be bought
  - Do not say ERP! You could however name ERP *modules* you know provide the required use cases
  
- ▶ Or scope applications to be built
  - “If you end up identifying a hundred use cases, either your system should be broken down into a number of smaller projects, or you are modelling use case pathways rather than use cases.
  - If you end up with one use case, then you need to move down a level of abstraction or get a bigger project.”
  - After Ivar Jacobsen





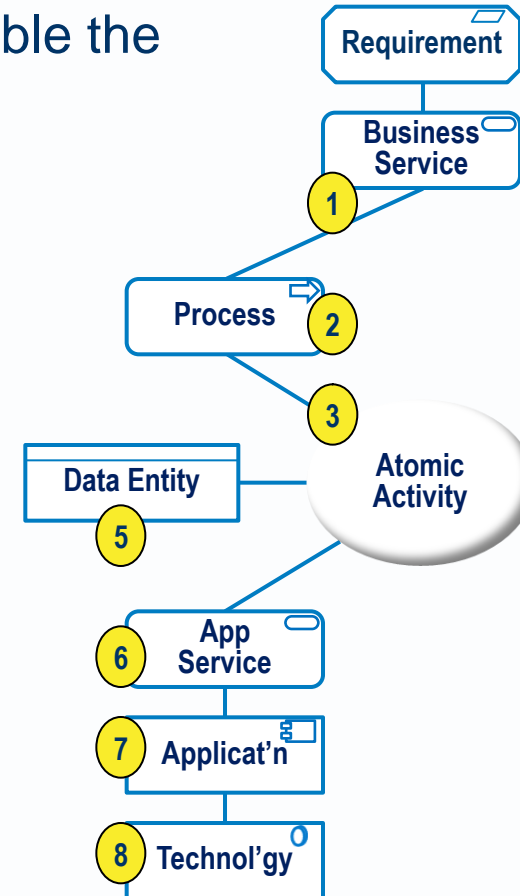
# Example tailored from one in TOGAF 8



Precondition: Sales visit to customer premises has been agreed and scheduled	Human actors (roles)		Computer actors (roles)	
Process	Customer	Sales person	Client app use case	Data centre app
1 Initiate sales process with the customer	Accept sales person	Greet customer		
2 Discuss customer requirements	Explain problems & requirements	Listen		
3 Work with customer to create a product configuration	Select one option based on capabilities	Show product and configuration options	Get product description Assemble configuration	Product Configurator App
4 Verify that the desired configuration can be delivered		Show availability to customer	Check configuration availability	Inventory App
	Accept	Show delivery date to customer	Get delivery date	Scheduling App
5 Determine price of requested configuration	Accept	Accesses price system and Shows price to customer	Price configuration	Pricing App
6 Confirm customer desire to purchase	Accept	Show offer		
7 Place an order		Captures order details and prints out fax	Enter order Get fax response	Order App
8 Capture customer signature	Sign	Requests signature		
Post condition: Order captured				

## 8. Define Infrastructure

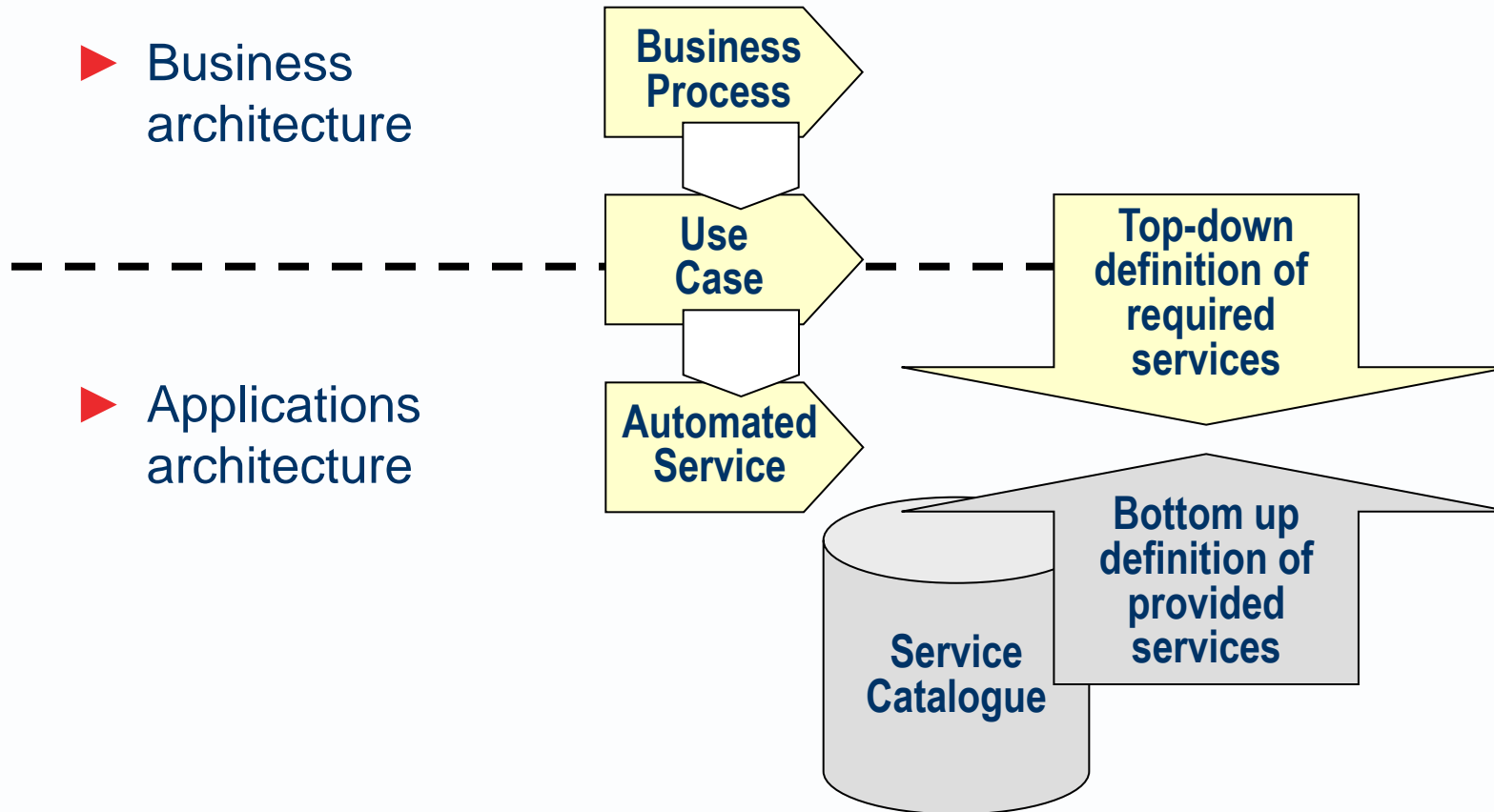
- ▶ List Infrastructure Technology components needed to enable the Applications
  - Platform technologies
  - Networks
  - Etc.



## When to stop?

- ▶ “The level and rigor of decomposition needed varies”
- ▶ If a process step is coarse-grained and not defined in sufficient detail for application procurers or designers
- ▶ And there remains more time, budget and resources for analysis
- ▶ Then divide that process step into shorter process steps and cycle through the earlier analysis process
- ▶ Else, stop the analysis

# Decomposition from business process to application process



# Where does business architecture stop and data and/or application architecture start?

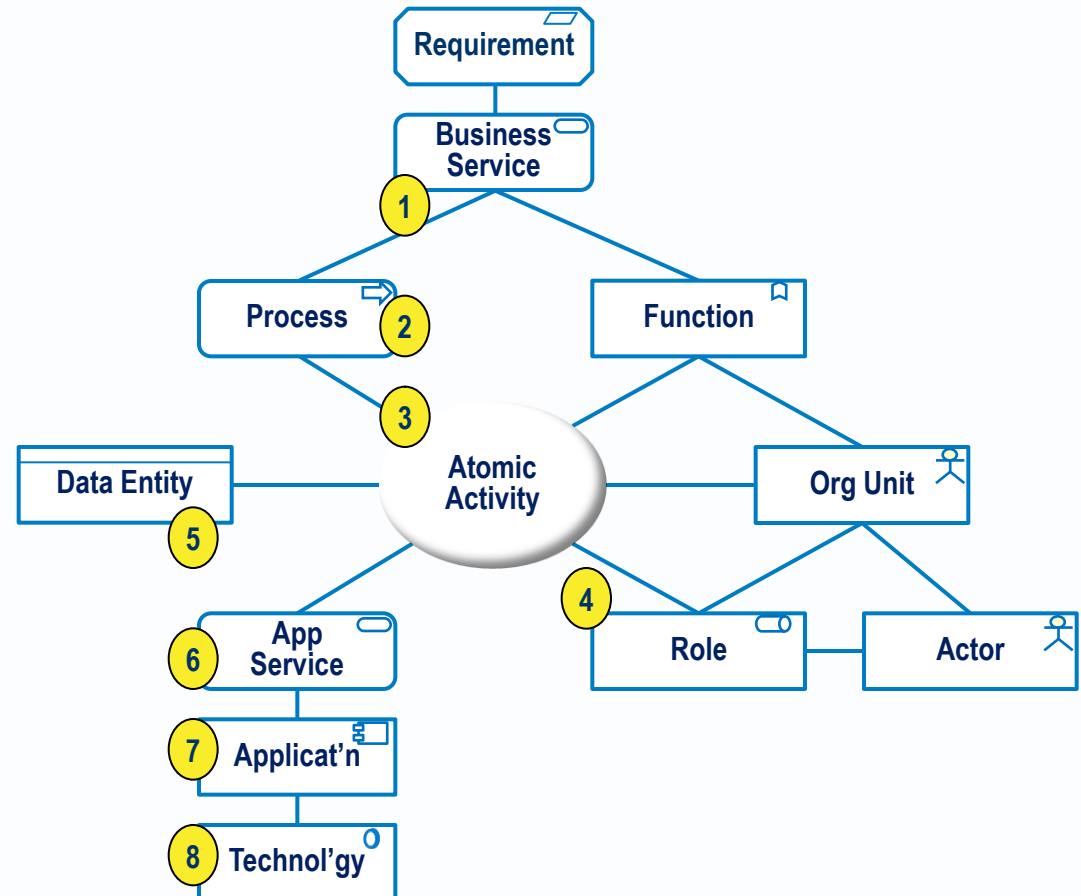
- ▶ A human process is typically decomposed to the level where an OPOPOT process at the human computer interface (HCI) is definable in an application use case.
- ▶ Consider the process called “calculate insurance premium”
- ▶ This might be documented
  - a pre or post condition of a business process step
  - a derivation rule for a business data item
  - a pre or post condition of an application service or use case
- ▶ *You* decide which architecture domain to document the process in

# Scenario-driven analysis and design (RECAP)

1. Identify required Business Service
2. Encapsulate Process as a Service
3. Decompose to OPOPOT activities
4. Assign activities to Roles
5. Define Data created and used
6. Define App Services needed
7. Define Applications
8. Define Infrastructure

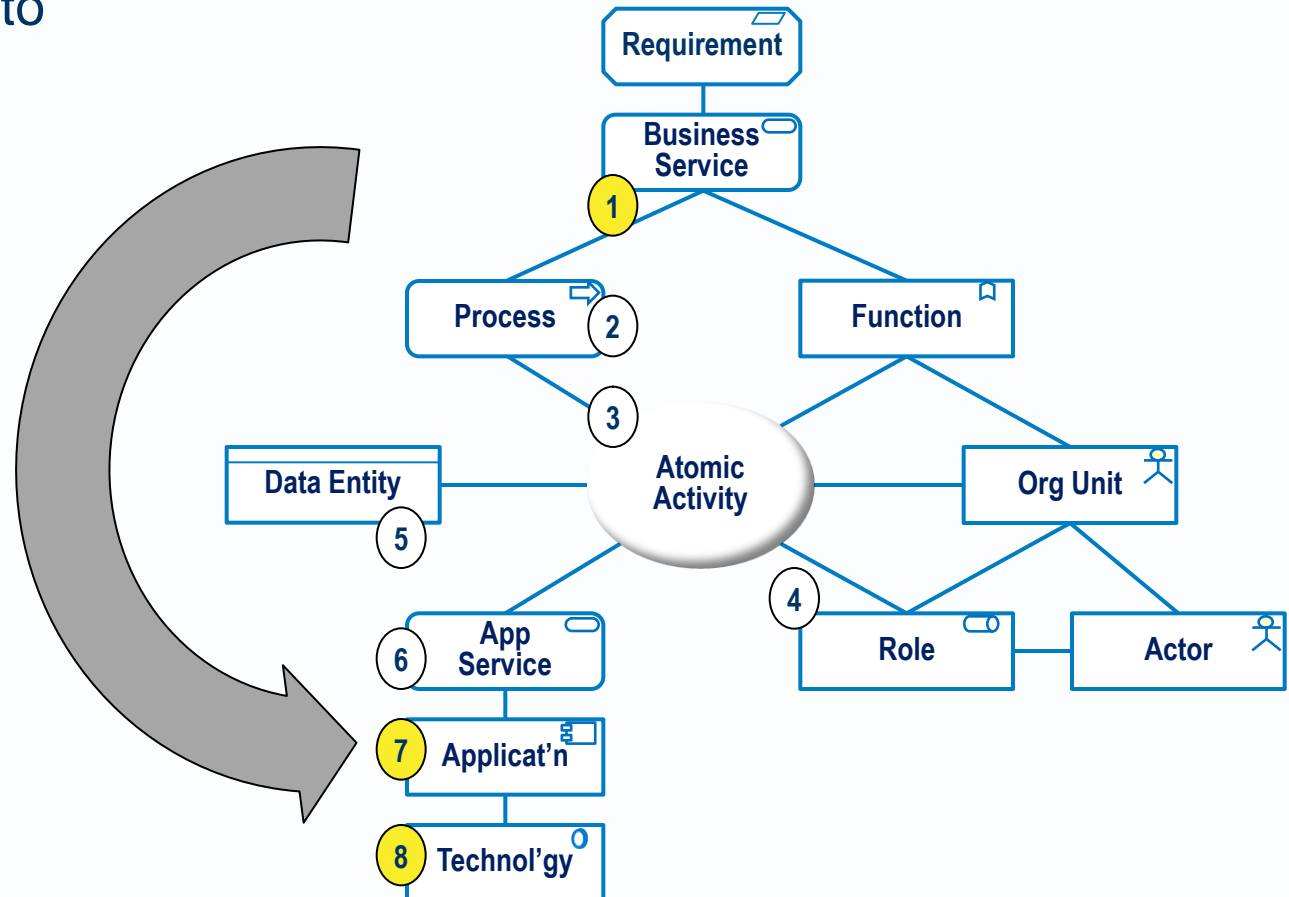
## Note:

- ▶ Business before technology
- ▶ External before internal
- ▶ Behaviour before structure
- ▶ “The level and rigor of decomposition needed varies”



# Scenario-driven analysis and design

► Beware: managers like to short cut the analysis!



# Principle: assign structures to perform required behaviours

## ► Define required behaviour

- Uncover requirements thru' process decomposition:
- What does the business do? What do people do?
- What must be automated or supported by creating and using data?
- Define end-to-end processes which are triggered by events and produce results

## ► Design the structure

- Design the capability
- Considering non-functional requirements such as flexibility and performance
- Define actors/components needed to perform the processes

