Extras for "Concept Modelling for Business Analysts" & "The Data-Process Connection"

- *Essentials of Business Processes*
- *Essentials of Use Cases & Services*

Prepared for SVB Sociale Verzekeringsbank 29 november 2024

Alec Sharp
Senior Consultant, Clariteq Systems Consulting Ltd.
West Vancouver, BC, Canada
asharp@clariteq.com
www.clariteq.com

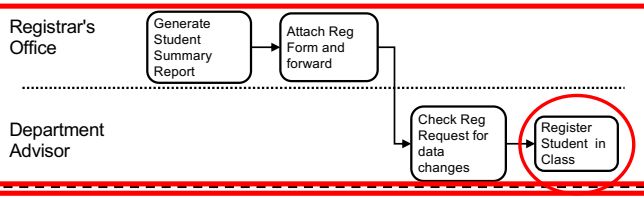
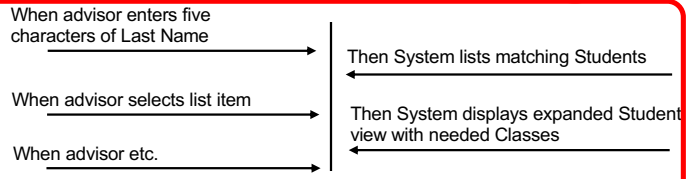
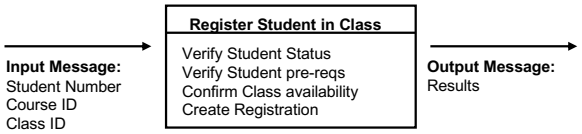
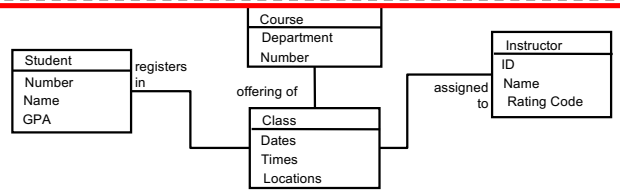
CLARITEQ

S V B voor het leven Sociale Verzekeringsbank

AdeptEvents

Essentials of Business Processes

Business Process – a vital perspective for Business Analysis

Framework Layer		Technique sample	What it covers
Goals	*Business Objectives*	The university is initiating the "Strategic Enrollment" program to raise Student graduation rates in part by ensuring Classes are available for Student registration when needed.	✓ ***Project Charter*** – documents the rationale, objectives, scope, and success measures for the project
Process	*Business Process*		✓ ***Process Model*** - shows "what" in a Scope Model, then "who & how" in a Workflow Model – the steps done by the actors in the process
Application	*Presentation Services (user interface)*		✓ ***Use Case*** – models how an actor interacts with a system to obtain (trigger) a service, typically to complete a step in a process
Application	*Business Services (rules & logic)*		✓ ***Service Specification*** - describes a service – a package of rules and logic – that is triggered complete or respond to a business event
Data	*Data Mgmt. Services (databases)*		✓ ***Business Object Model*** - depicts the things and the facts about things the organisation needs to record; the things (Business Objects) are what processes and solutions act on.

Business Process: a great framework for *Business Analysis*

Use Cases and Services: where we capture *Functional Requirements*

Business Object Model: a great platform for *Business Analysis*

Business Process themes and overview...

Three main themes:

1. *Simple* techniques, *rigorously* applied, help us achieve *more* in *less time*.
2. *Communication with* and *engagement by* the people who *do the work* through the use of simple, consistent techniques.
3. *Not* just a technocratic undertaking – we take a *holistic* view that includes *human*, *social*, and *organisational* factors.

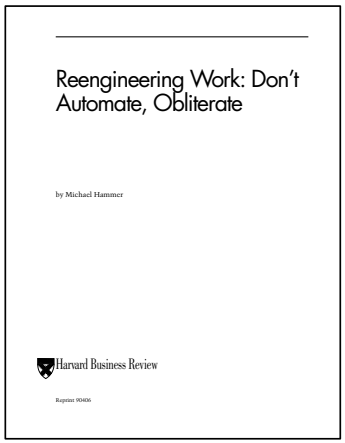
First, the fundamentals

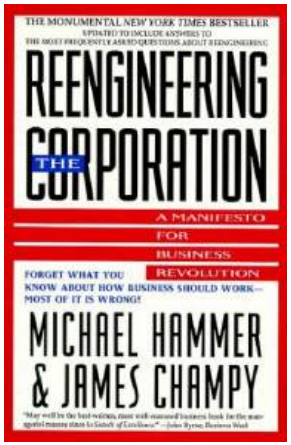
- Some things you need to know about *business processes*
- A three-phase methodology for *Business Process Change*

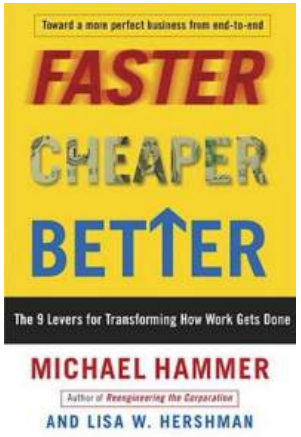
1. Confusion – what **is** a "business process?"

1. It is essential to have clarity on what a *business process* really is
2. Performance measures may be *functionally aligned* - work *against* business processes
3. Success with business processes requires a *holistic view* in which *six enablers* are considered
4. A business process can't be great at everything – a single *differentiator* must be chosen

In the early 1990s, Michael Hammer popularised the focus on *business process*







Introduced core terminology:

- end-to-end, cross-functional, functional silo, ...
- even *business process*

Still, people and organisations miss the point...

Lesson #1 – Never assume everyone agrees what a "process" is

We need some help with our *Product Lifecycle Management* process.

I spend all day writing business processes, like the process to *Revise Product Brochure Image.*

Not a single process – it's a *family* of multiple business processes (a *process area* or *process domain*)

Not an entire process – it's a *procedure* providing instructions for a single task (SWI – standard work instructions)

A whole *spectrum* of interpretations of *process.*

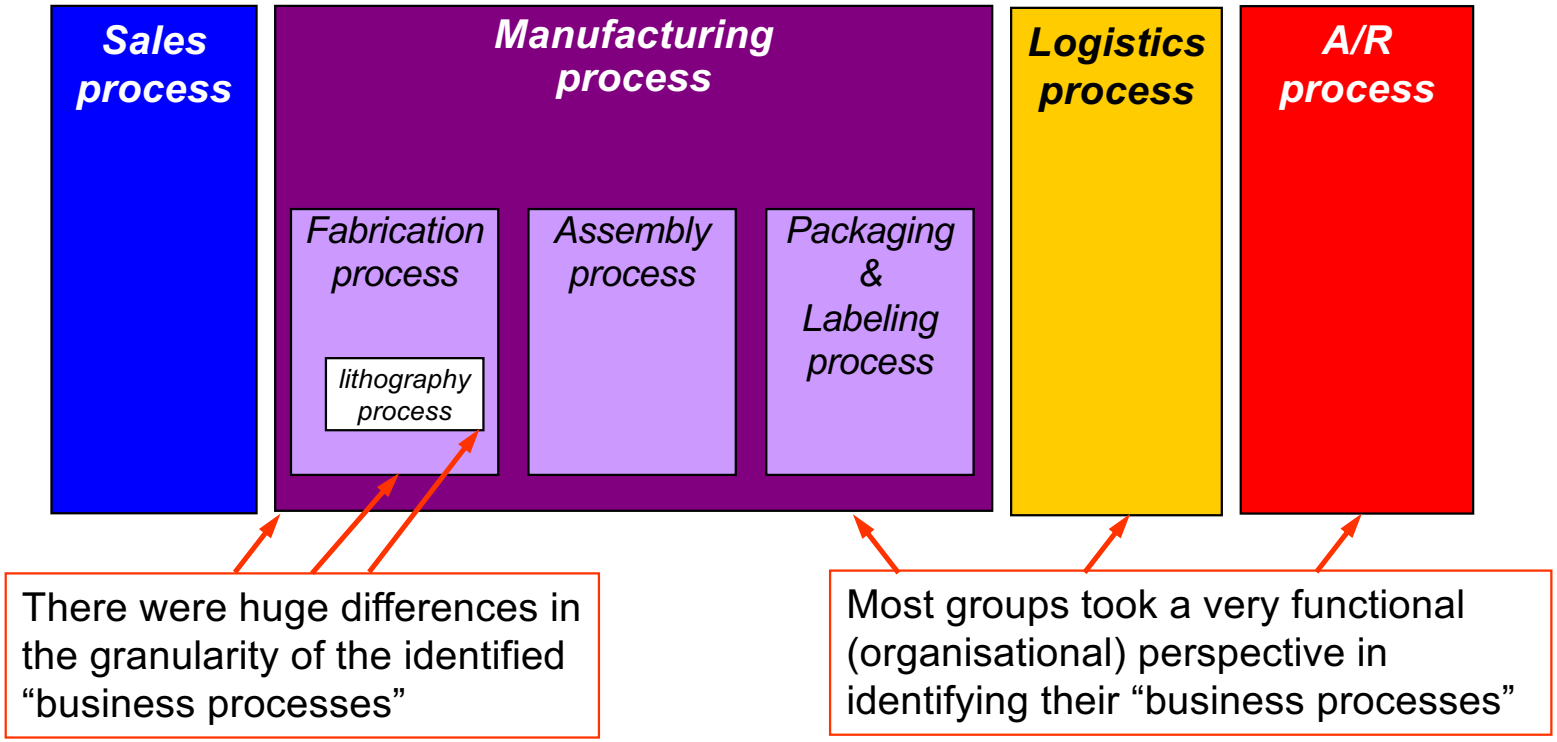
Seek balance – a "business process" lies between the extremes

Most people hear *process* and think *procedure!*

The key issues – granularity and orientation

A real life (and expensive!) example

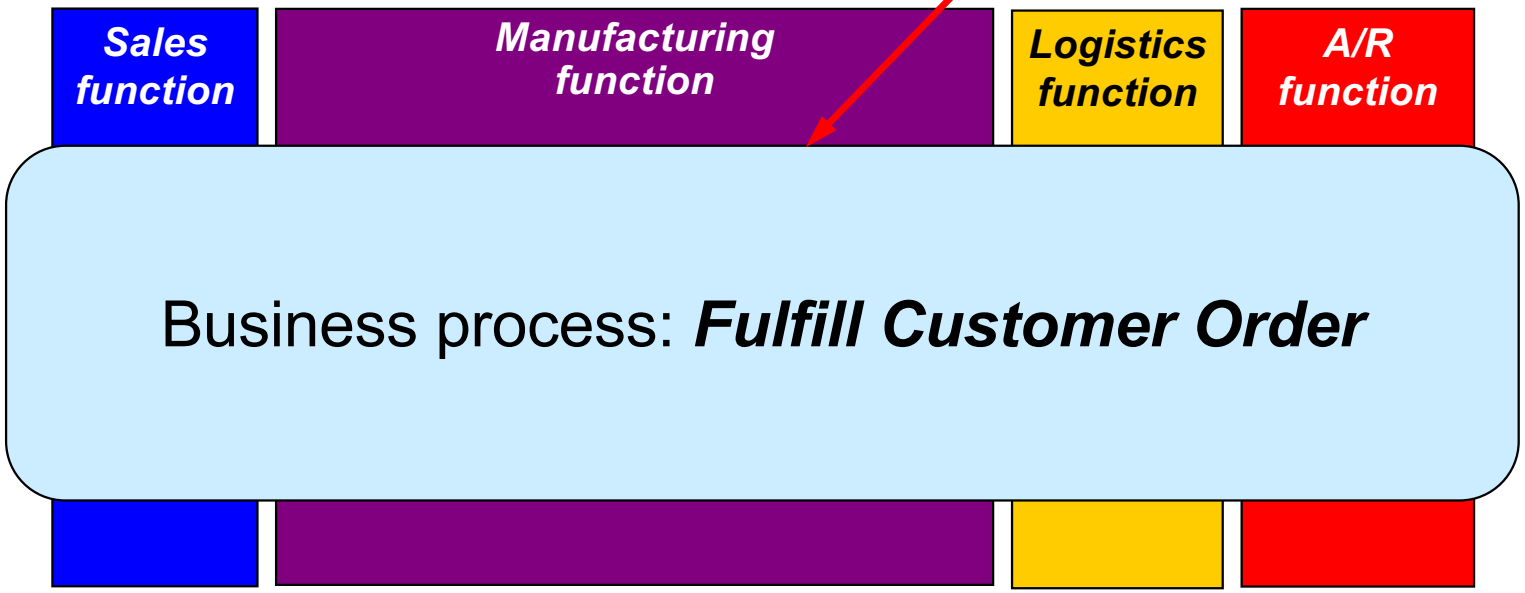
As part of a massive system implementation, a global manufacturer identified the *business processes* that were expected to improve:



The problem? *These aren't processes – they're functions!*

The "real" business processes were missed

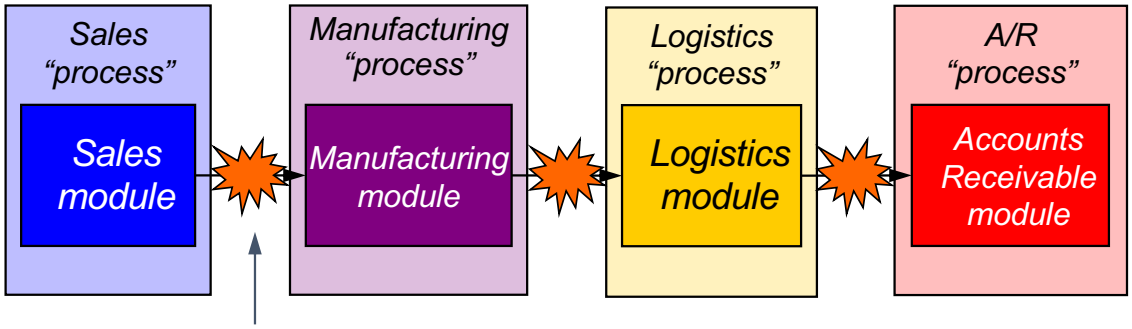
Everyone confused "process" and "function." None of the actual end-to-end processes were correctly identified.



"Business Process" =
end-to-end, cross-functional, business process.
"Larger" than people think – from *initial* trigger to *final* results

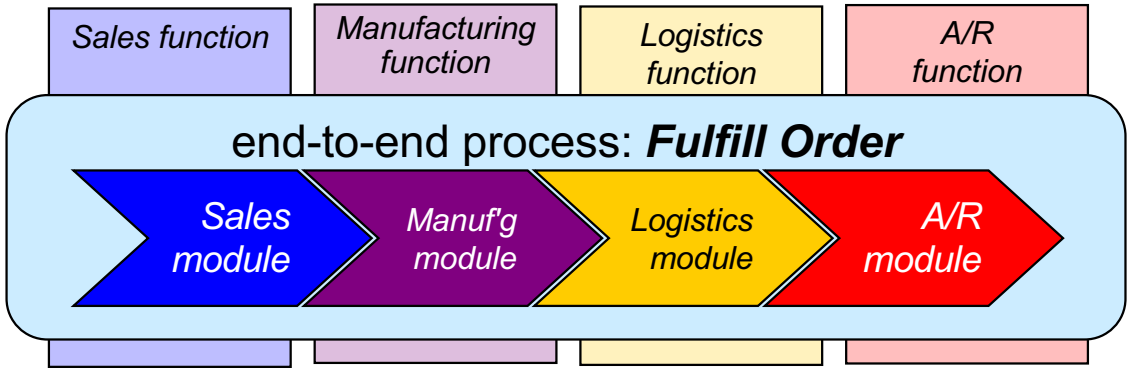
Impact of confusing function and process

Implementing SAP without clarity on "process":



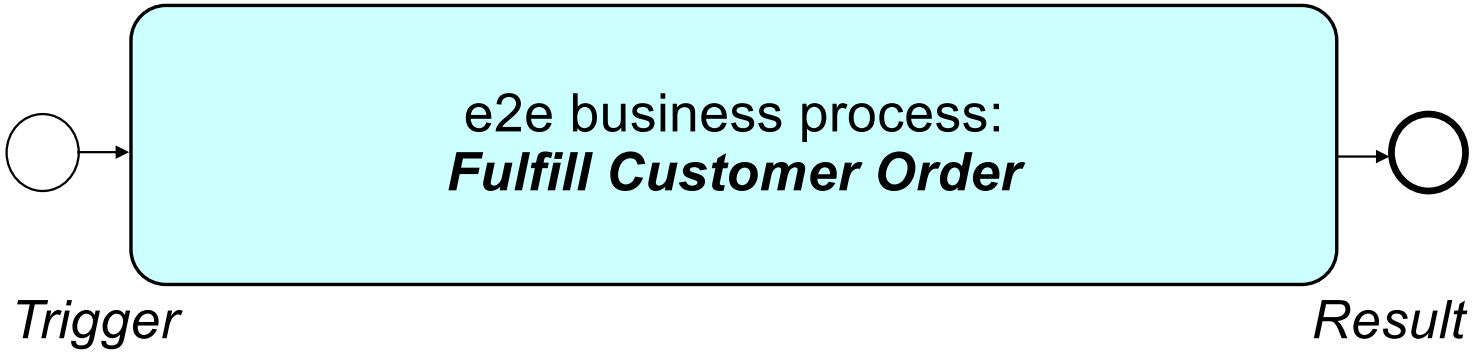
Conflicts: timing, coding, terminology, data formats, *performance targets*, …

SAP re-implemented in a process-driven configuration:

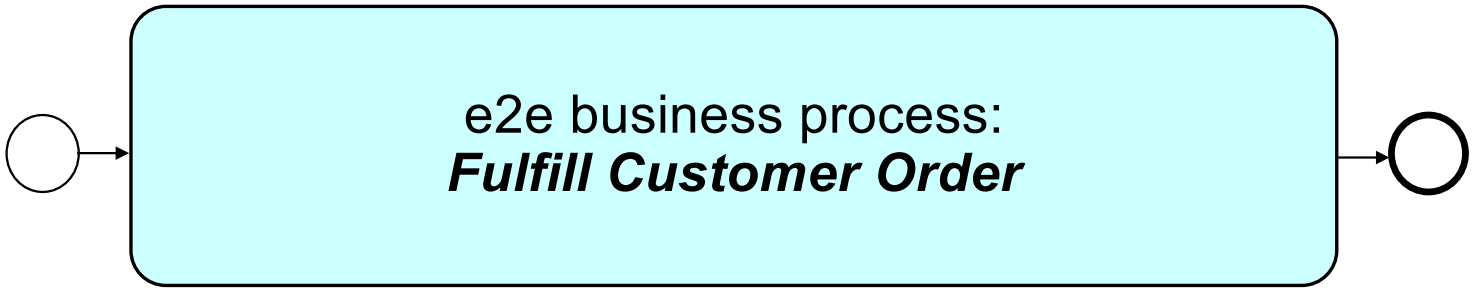


Same software, radically different outcomes

Discuss - what are the boundaries of the process?



What are the boundaries of the process?



Trigger

Order received? *No.*

Before that…

- Contract is Finalised
- Price & Schedule are Negotiated
- Specifications are Confirmed

And before that…

- Demand is Signalled. *Yes.*

Result

Order is Shipped? *No.*

Order is Received? *No.*

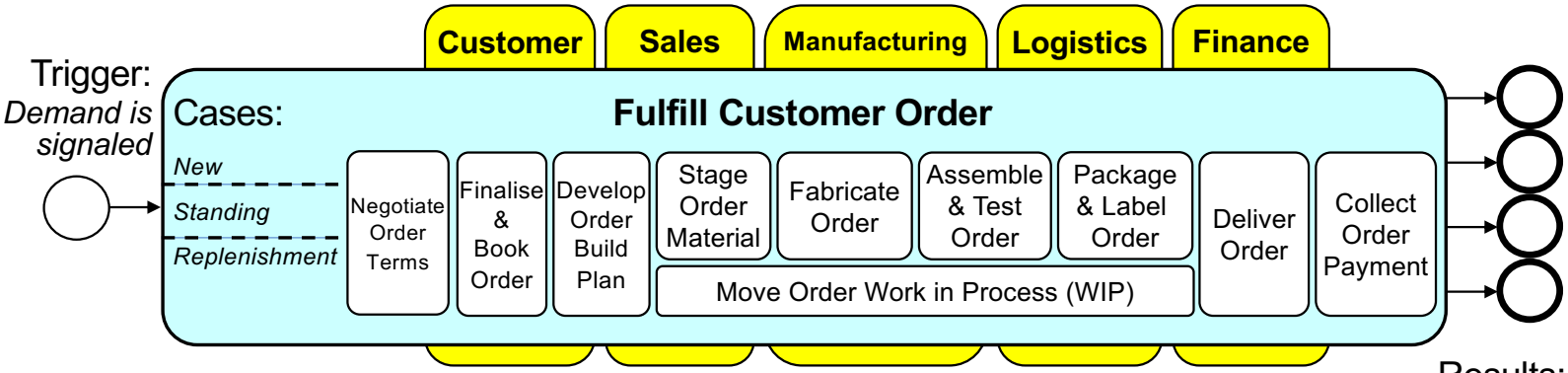
Order is Received, Tested, and Accepted? *Yes.*

Any other results?
Yes, for other stakeholders.

Always trace to the *earliest* trigger,
and to the *final* results for *each* stakeholder.

Process Scope Model – "what" first, "who and how" later

I build a *Process Scope Model* & a *Process Summary Chart* on ~100% of *Project Recovery* assignments -



"TRAC" –

*1 – **T**riggering event or events*

*2 – **R**esults: final outputs*

- result(s) received by the process' primary customer
- result(s) for other stakeholders (performers, owner, supplier, regulator, …)

*3 – **A**ctivities: 7 +/- 2 phases, milestones, or sub-processes*

- a phase achieves a significant intermediate result
- simply ask the participants for ~5 to 7 milestones within the process

*4 – **C**ases*

- main variations, e.g. "*new* order" vs. "*standing* order"
- verb – *qualifier* – noun

essence** of the process **("what")

5 – Functions or Organisation Units

6 – Actors and responsibilities

7 – Systems, data sources, other mechanisms

***as-is elements** of the process, for clarification **("who and how")** (6 and 7 not shown)*

Results:

Customer:
Goods received, tested, & accepted

Owner:
Payment received

Performer:
Commission credited

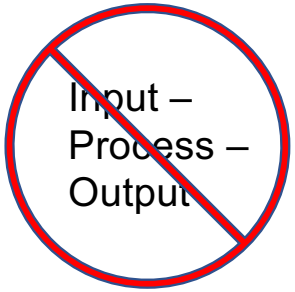
Industry Association:
Order stats reported

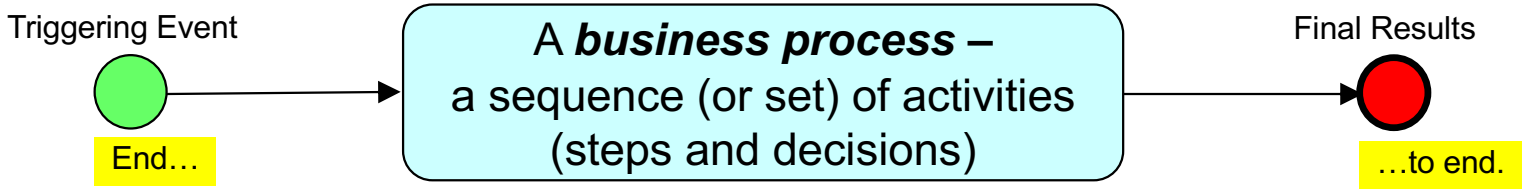
Always construct a *Process Scope Model* & a *Process Summary Chart* before diving into Workflow Modelling / Swimlane Diagramming

The essential framework

Business Process:

- a sequence (or set) of *activities* (steps and decisions,)
- initiated in response to a *triggering event*,
- that achieves a defined *result* for each process stakeholder





- Three types of events:
  - Decision-based (action)
  - Time-based (temporal)
  - Data-based (conditional)
- The *earliest* triggering event

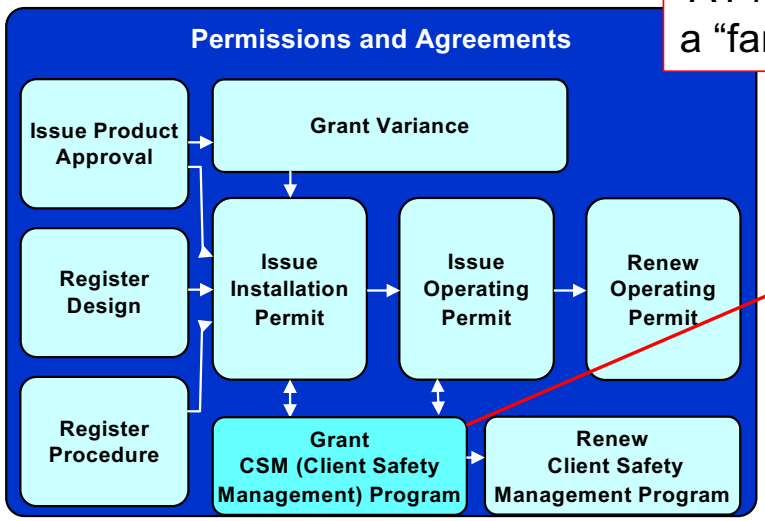
- Important processes are virtually always cross-functional and involve multiple actors / roles
- May be a defined *sequence*, or a more ad hoc *set* of activities
- First, identify *"what"* it includes – Trigger, Results, Activities, Cases ("TRAC")
- Later, we add *"who and how,"* then map the process flow, if there is one

- Three types of results:
  - A service
  - A good
  - Information
- The *final* result

"What" before diving into the "who and how"

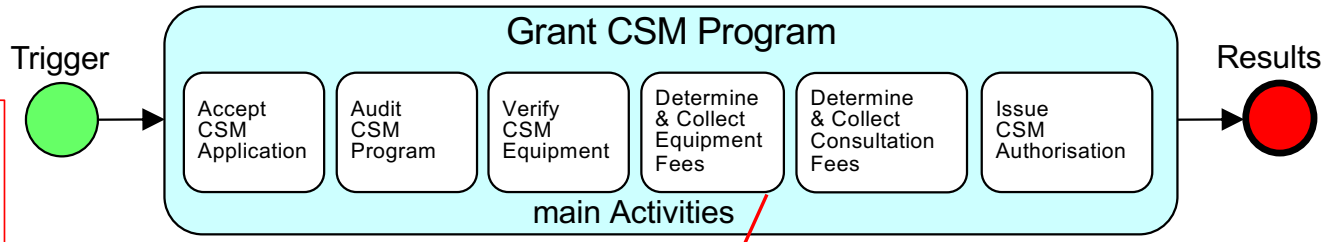
Taxonomy: a collection of processes vs. a process vs. a procedure

A *Process Area* or *Process Domain* –
a "family" of related Business Processes:



An *end-to-end process* – "Grant CSM Program,"
from *application* to *authorisation,*
involving many departments, external organisations,
participants, and procedures.

Business Process Scope Model (TRAC) – pure "what"…



Cases: New, Legacied, etc.

"how" to complete a task…

Business Process:
A sequence or set of *activities*
that delivers significant *results*
for the process' customer
and other *stakeholders*

Procedure:
A set of step-by-step work instructions
(a job aid) for a specific task or activity
that will yield identical results every time

Procedure – Calculate Unit Registration Fees:
For each Unit:

- Determine Unit Type and Unit Risk Factor;
- Apply Registration Fee from Reg. Fee Table;
- Identify additional Inspection fees from…

Naming conventions will make life easier

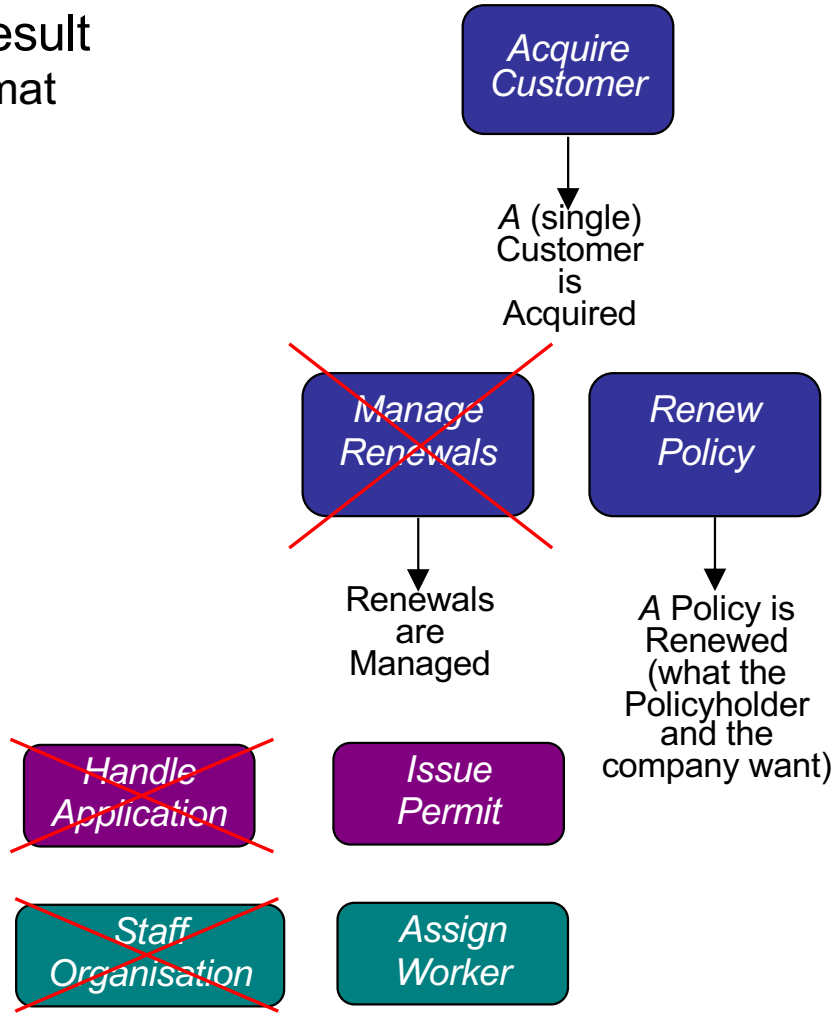
1. The process name ***must*** indicate the expected result
   - Name potential process in "active verb – noun" format
   - Restate that name as a result ("noun is verbed")
   - Ensure this is the intended result of the process: *discrete*, so results are *identifiable* & *countable*

   - ***No mushy verbs:*** manage, monitor, administer, handle, track, support, maintain, etc.
   - ***Active verbs only:*** *Evaluate* Prospect, *Acquire* Customer, *Fill* Customer Order, *Resolve* Customer Issue, …

   - Applies to business processes, phases (subprocesses,) activities, steps, …

2. Name process from customer's perspective (what do they want from the process?)

3. Name process in the singular



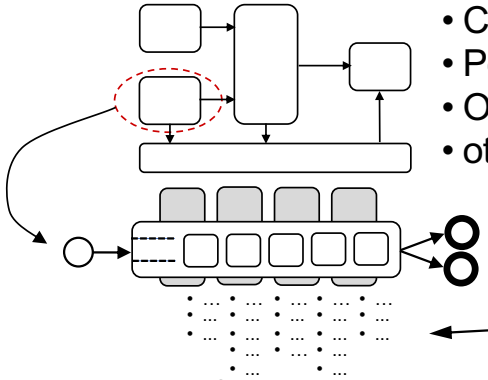
Our three-phase methodology – proven, practical, & agile

Some goal or issue, not rigorously specified

1 Establish Process Scope and Objectives

Identify & scope the process with a Scope Model & a Process Summary Chart; Optional - build a Concept Model

Complete ***initial*** as-is process assessment, and to-be objective setting, by ***stakeholder***



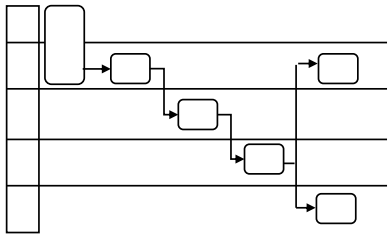
- Customer
- Performers
- Owner
- others…

- ID processes & draw *Process Landscape* (Optional – only if you have a large scope)
- ID **T**rigger, **R**esults, main **A**ctivities, **C**ases (**TRAC**) & draw *Process Scope Model* – focus on *what, no* reference to *who* or *how*
- ID involved functions & mechanisms (*who* and *how*) & draw *Process Summary Chart*
- Conduct *stakeholder-based assessment*

2 *Understand* the As-Is Process

Perform more ***detailed*** as-is modelling: an Augmented Scope Model & optionally, Workflow Models

Complete ***final*** as-is process assessment by ***enabler***, and generate to-be improvement ideas



- Process
- IT
- M&M
- HR
- P&R
- Fac. or…

- Develop *as-is* models:
  - *Augmented Scope Model* – add ~5 – 7 more detailed Activities for each main Activity
  - (Optional) as-is *Workflow Models* – *only* enough detail to understand process behaviour
- Conduct *enabler-based assessment* and identify *potential improvements*

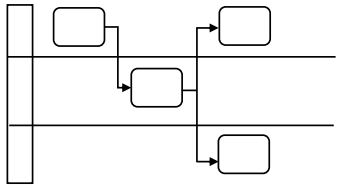
3 Design the To-Be Process

Refine to-be improvement ideas and determine 5-10 ***key features*** of the to-be process

Assess each to-be feature ***by enabler*** to ensure the new process is implementable and sustainable

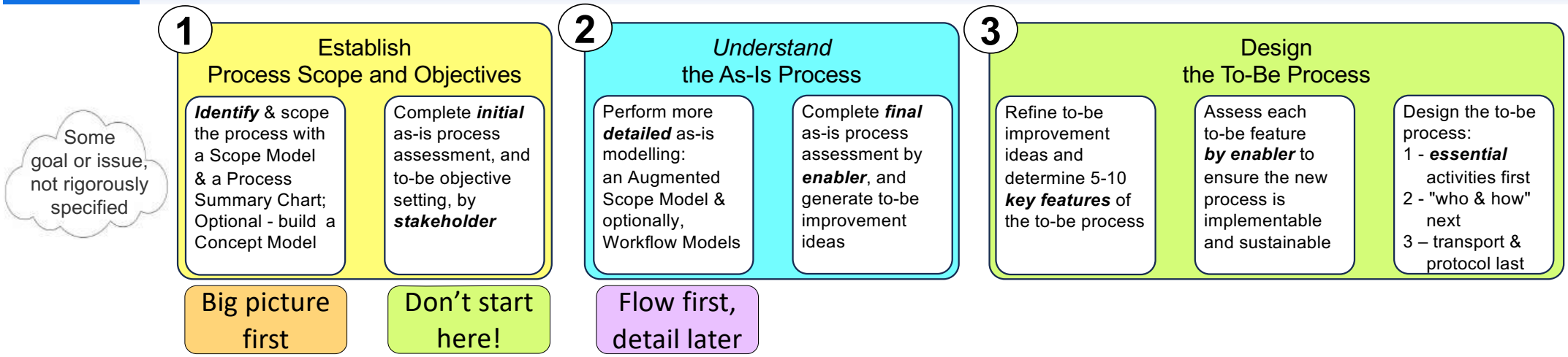
Design the to-be process:
1 - ***essential*** activities first
2 - "who & how" next
3 – transport & protocol last

- Select key to-be *Features*



- Assess each key *Feature* by enabler
  - Identify and sequence *essential* activities
  - Develop *Workflow Models* for essential activities by adding *who* and *how*
  - *…on to requirements definition and implementation*

Our methodology – three responses to three common difficulties



My *hardest* assignments

1 – Premature diagnosis of the situation

2 – Failure to identify true end-to-end processes

3 – A rapid descent into unhelpful detail

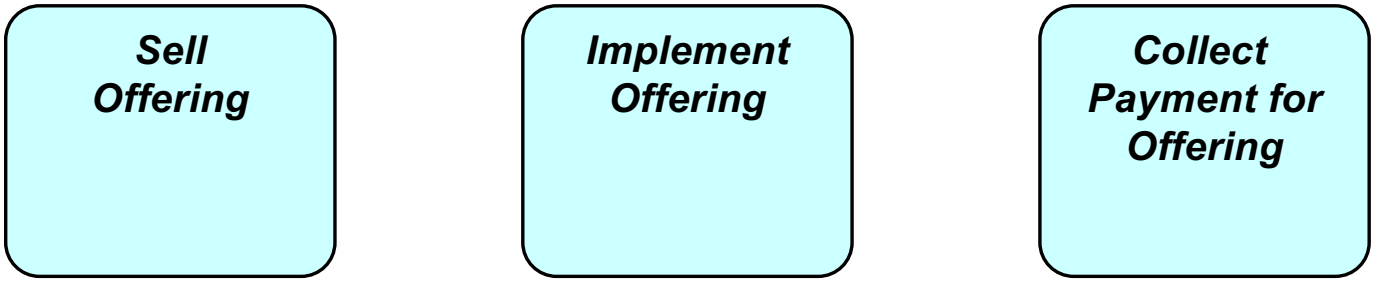
Don't start with a problem statement!
There will be some goal or issue, but don't formalise it ***yet***.
And remember… it may not be a "process" issue.

Rigorous techniques to identify real business processes – a Process Scope Model and a Process Summary Chart make scope and context visible.

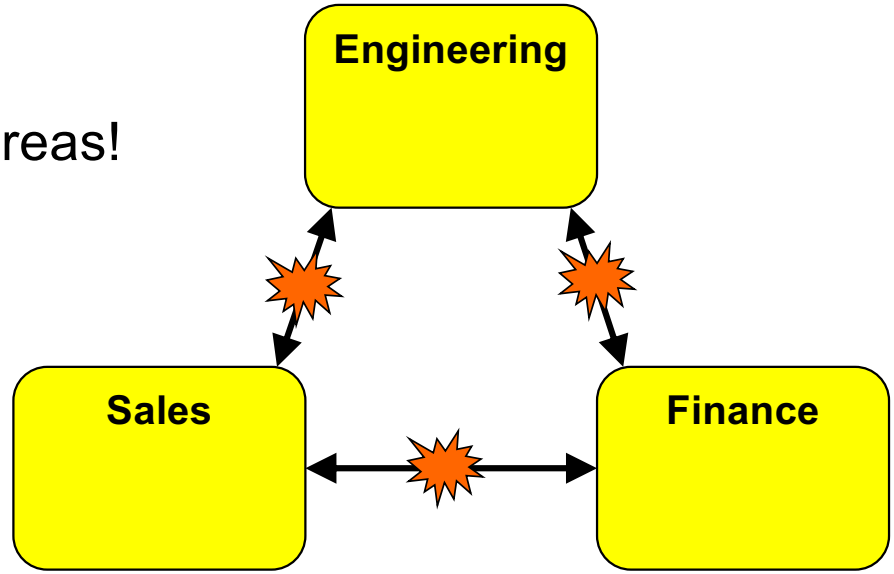
Clarify the big picture, then take a *controlled* descent with well-defined levels of detail.

Another Business Process example, if we have time

A regional telecommunications provider (the "Telco") thought they had three main Business Processes, and efforts to improve them were failing:

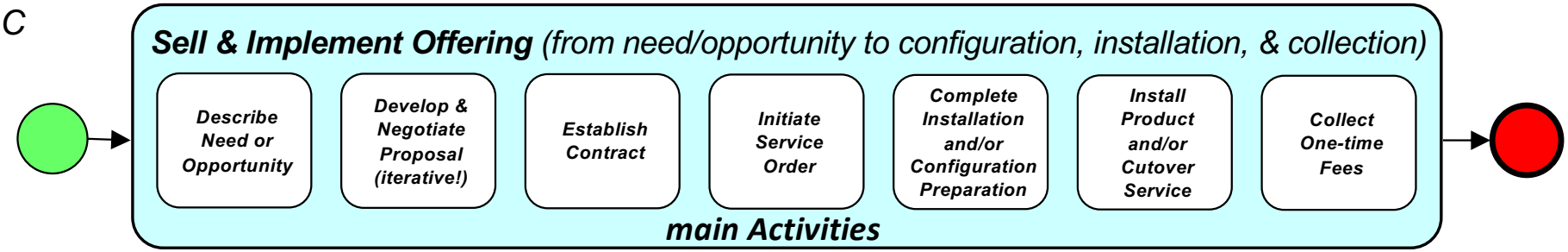


The outcome…
conflict between functional areas!



Process Scope Model showed ONE process not THREE

TRAC



Triggering Event:
- *Prospect / Customer* expresses *need*
- *Telco (Inside Sales, Marketing, Sales Rep, …)* recognizes *opportunity*

Cases:
- BU with or without Telco Internet, no cabling *(our focus)*
- initial installation
- service only
- product only
- mixed

Other factors:
- TBD

Results:
Customer:
Product / Service is *installed and operational* per original or amended contract terms
Telco:
- Ongoing source of *revenue* in place
- One-time *fees* collected

Employee:
- *Commission* or *referral credit*

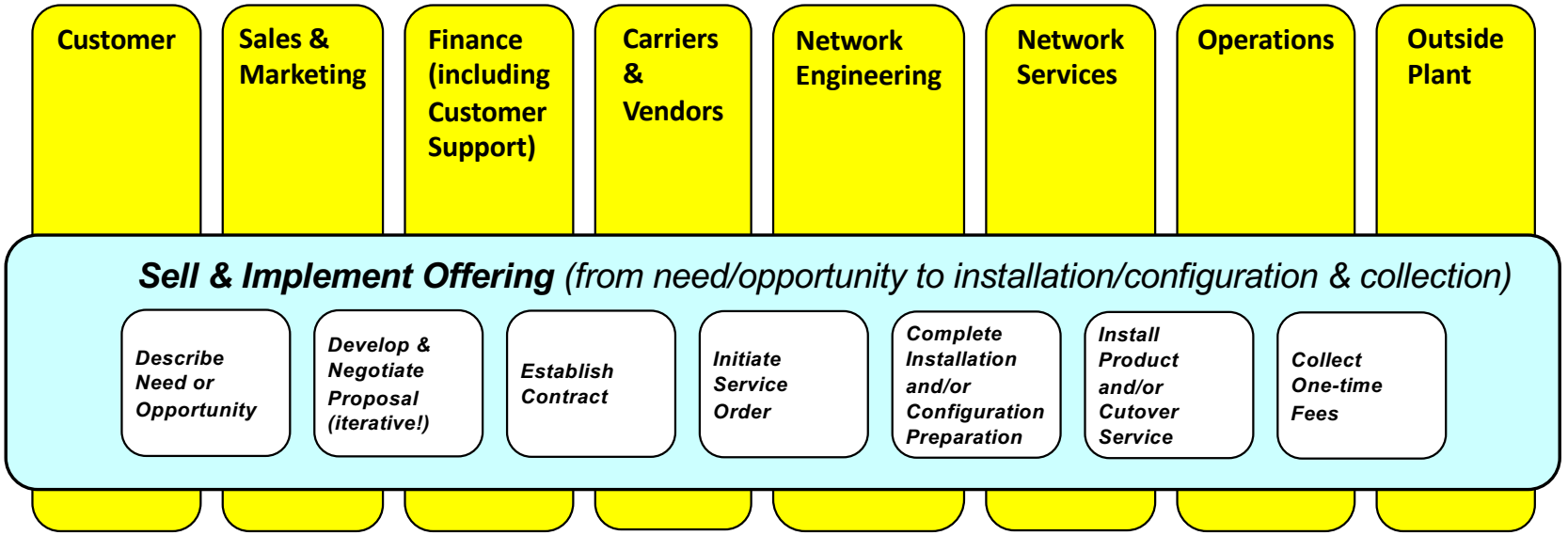
Agent:
- *Commission*

The "token," a Service Order, is changing state from *need/opportunity* to *configured, installed, & collected*.

The Business Process could be named "Fulfill Service Order" but the client wanted to name it "Sell & Implement Offering."

- President reports *culture change*. "We're all in this together!"
- An end-to-end, cross-functional Business Process is a great lens to view *organisation conflict and disfunction!*

Process Summary Chart – my favourite diagram!



Process Summary Chart (a.k.a. ”Process vs. Function Chart”) adds “who” at the organisational unit or functional level.

Nothing else clarifies "Process" vs. "Function/Organisation" as well.

Great for putting details of Activities or Functions in context, e.g. …

Multiple roles by organisation for "Sell & Implement Offering"

Customer

Roles:

- Office manager or Owner (Smaller)
- IT (Larger)
- C-level (CIO, COO, CFO…)
- Third party IT vendor or agent
- Customer Project Coord.

Sales & Marketing

Roles:

- Senior. Account Execs
- Strategic Rel'nship Managers
- Account Rep 1
- Inside Sales Rep

Finance (including Customer Support)

Roles:

- Sales Admin
- Order Writer
- Billing Rep.
- Customer Support Rep.
- Director of Customer Support
- Receiving and Posting Payments (what role does this?)

Carriers & Vendors

Roles:

- Port Out Specialist (for CS Record) CSR/LSR
- IT Person
- Local government
- "Call before you dig"
- Customer Project Co-ord (int/ext consultants or phone vendors)

Network Engineering

Roles:

- System Admins (assign IP)

Network Services

Roles:

- BU Tech (survey)
- Switching Specialist (NS Spec)
- Network Services Coord / Provisioner

Operations

Roles:

- Sales Engineer
- CLEC Technician
- Material Manager
- Materials Specialist
- Project Manager
- Customer Training & Support
- Install Supervisor

Outside Plant

Roles:

- Drop Crew
- Lineman (not usually)
- Engineering Supervisor
- Outside Records Specialist

It was a shock to senior leadership to see how many roles were involved, often overlapping or unnecessarily

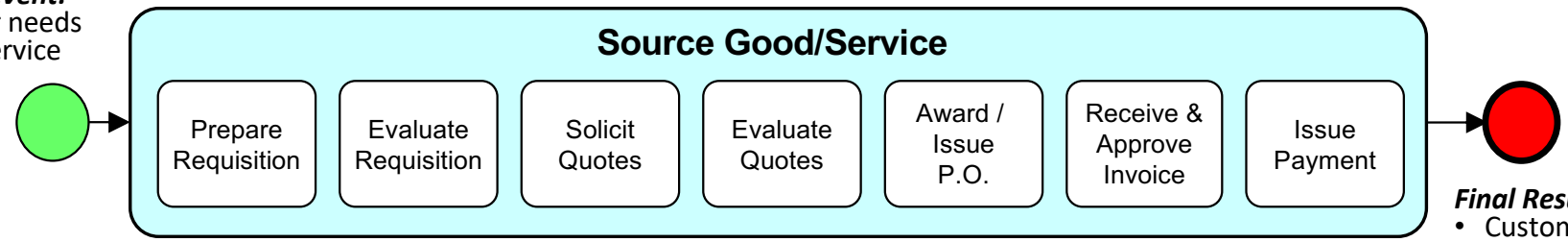
Another fast Augmented Scope Model example

Cases:
- $5000 - $25000 Goods
- $25000 - $50000 Goods
- $5000 - $25000 Services
- $25000 - $50000 Services

Assume everything <$5000 is purchased with a PCard

This example adds detail by major Activity (or subprocess/phase/milestone)

Triggering Event:
- Customer needs Good / Service



Final Results:
- Customer has received Good/Service:
- Vendor has been paid
  - via A/P
  - via PCard

Prepare Requisition

Develop scope of work / specs

Investigate potential vendors (and price?)

Solicit vendor quotes (just to get an idea)

Obtain approval (Department)

Verify Item and Account (General Accounting)

Submit requisition (visible to all)

Evaluate Requisition

Confirm completeness – get clarification this is actionable (scope sufficient)

Assign (or re-assign Buyer as necessary)

Identify MBE/SB opportunity (competitive) (co-op)
* sole source or co-op, vendor(s) known

Determine methodology
- sole source
- co-operative (piggyback on contract)
- competitive
- emergency

Solicit Quotes

Determine (additional) potential vendors

Solicit quote (including Bid Due Date)

Post quote (solicitation documents) in "the binder"

Resolve vendor queries

* Up to $200K, we control who gets solicitations; above, no control – it's "publicly advertised."

Over $200K there would be 20 more activities, and could be multiple award.

Evaluate Quotes

Receive quote (mail, fax, e-mail, …)

Confirm completeness

Verify suitable price, terms, and conditions (generally, low bid for equivalent)

Clarify (not negotiate) with vendor

Optional:
- Evaluate equivalency (for alternate)
- Confirm equivalency w. Customer

Identify vendor

Award / Issue P.O.

Generate Purchase Order

Notify Requestor

"Transmit / deliver" P.O.
* Pain point – we aren't sure when the vendor receives the P.O.

Issue Payment (Magic Happens Here)

* If multiple line items, different line items could go to different vendors;
* If multiple vendors, line items are not split.

Receive & Approve Invoice

Receive Good/Service
* Invoice could be attached

Accept Good/Service

Issue invoice (vendor)

Issue Payment

Receive invoice:
- from vendor
- from the department the vendor sent it to

* Vendor complains invoice is "lost"

If >$5000, match
- invoice
- PO
- receiver

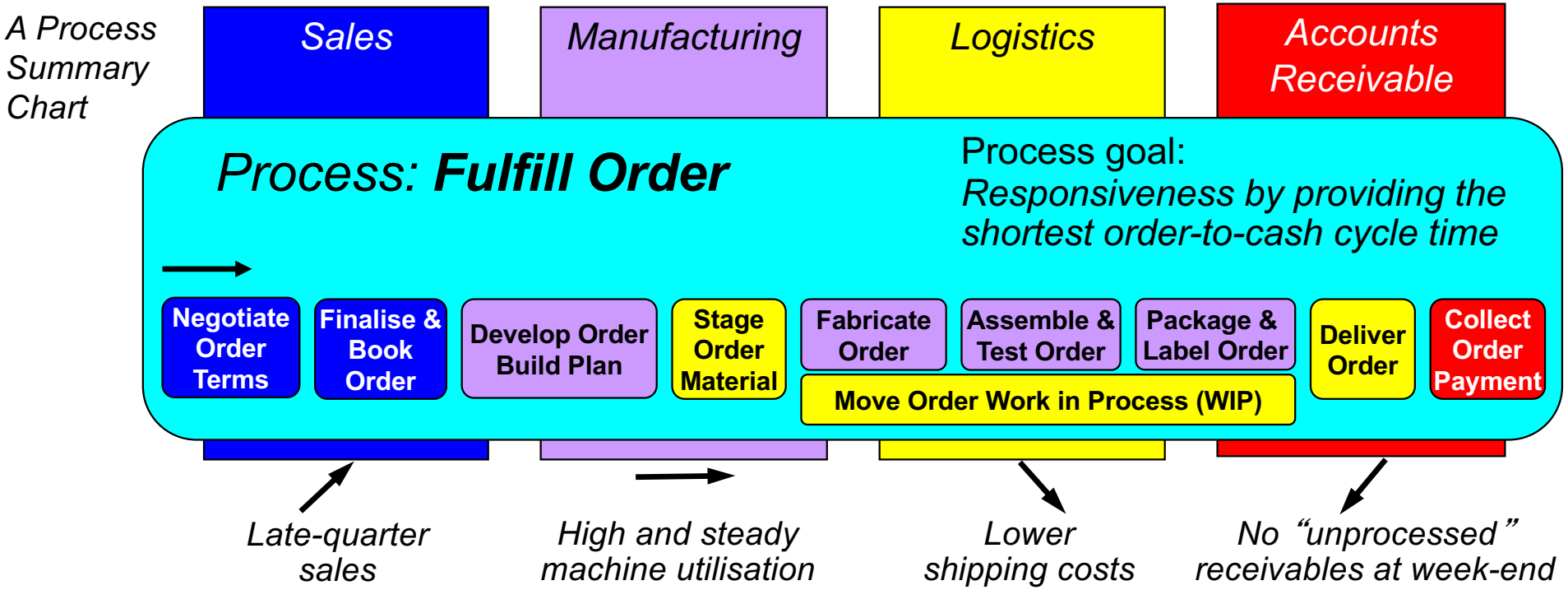
If <$5000, match
- invoice
- PO

* Could invoice $4K on $40K PO

Batch invoices for GAD

Receive payment

2. A common obstacle – misaligned performance measures

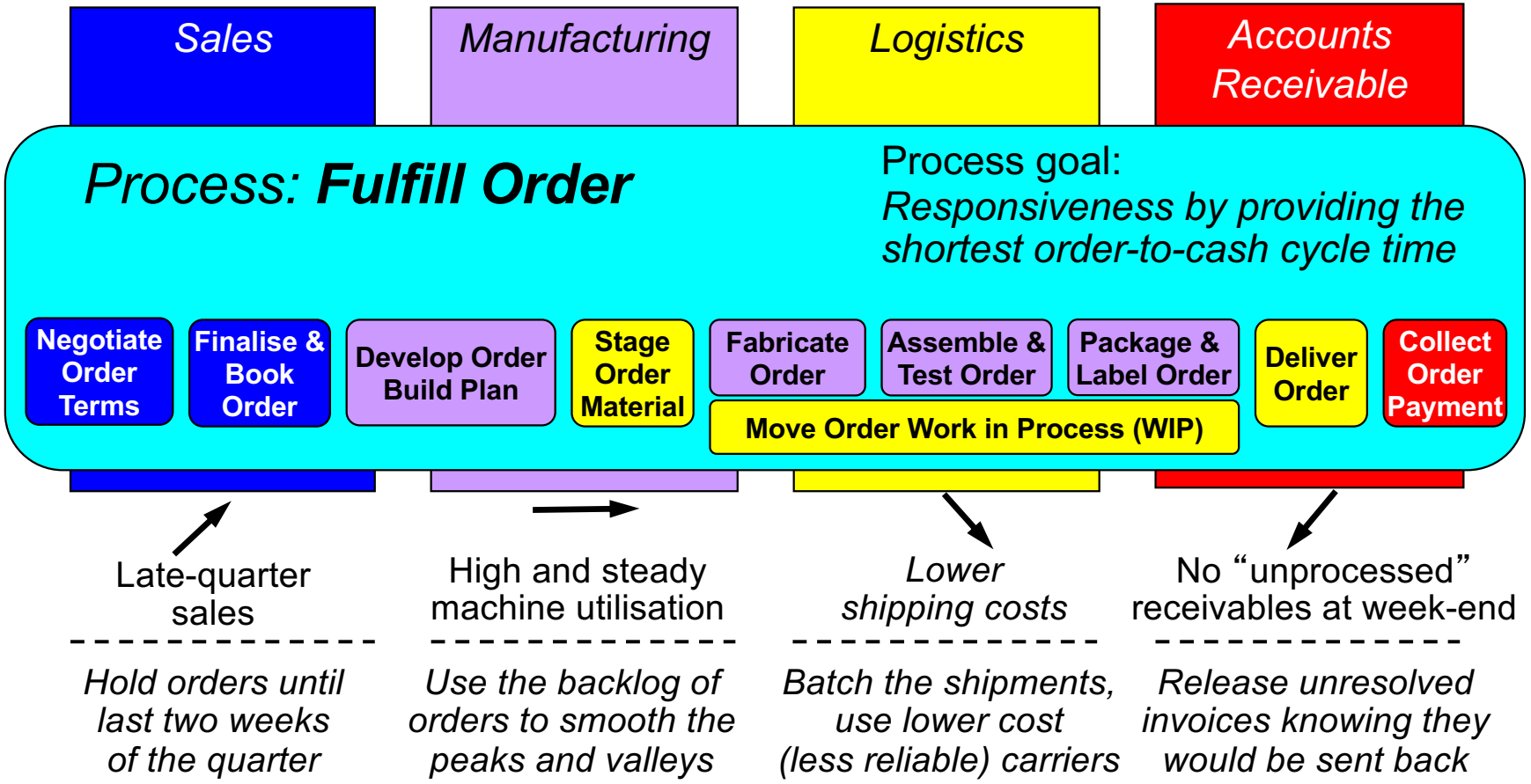


1. It is essential to have clarity on what a *business process* really is
2. Performance measures may be *functionally aligned* and work *against* business processes
3. Success with business processes requires a *holistic view* in which six *enablers* are considered
4. A business process can't be great at everything – a single *differentiator* must be chosen

But… performance measures were established *functionally, before* awareness of the *end-to-end process*

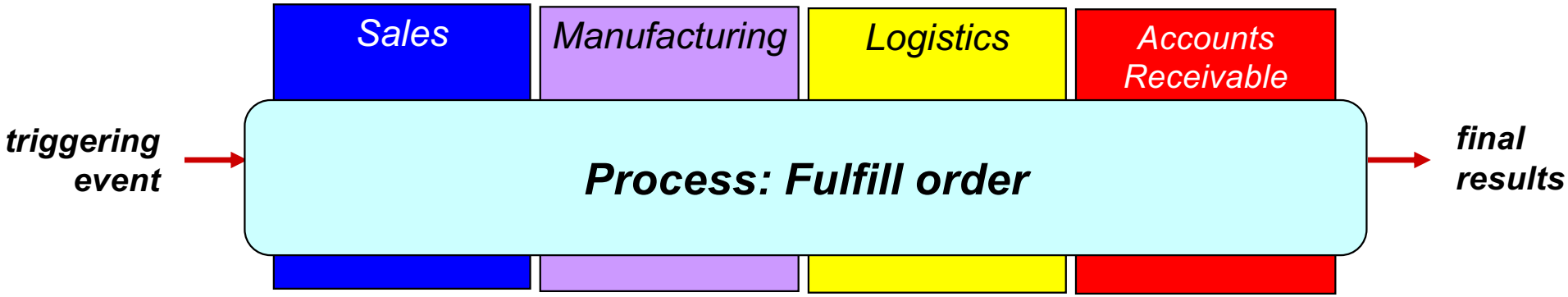
Think about it –
What are the likely impacts of these performance goals?
What will the different functions do to meet the targets?

Misaligned performance measures



Poor performance because each function was *working hard* to meet *uncoordinated, functional targets*

This doesn't mean functions are bad!



Process

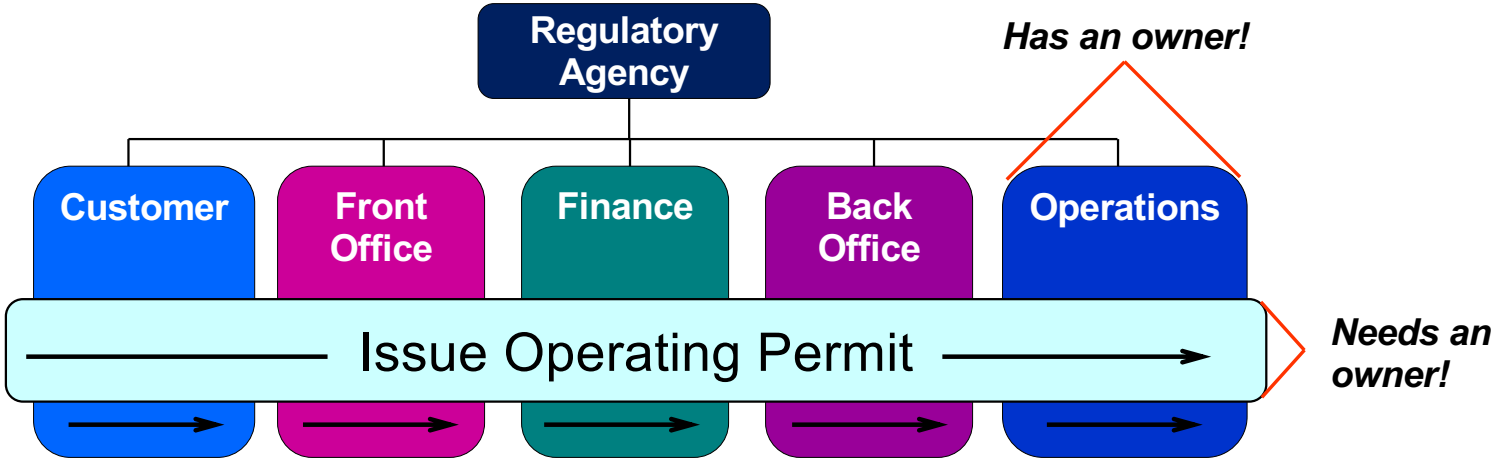
- End-to-end business processes deliver ***essential results*** by aligning the work of multiple functions
- Must be identified and managed as a whole

Function

- A centre of expertise – ***cylinder of excellence –*** an efficient way to provide resources across multiple processes
- Specialised skills, knowledge, tools
- Organisational design usually based on functional specialties
- We prefer not to use the somewhat negative term "functional silos"

Ultimately, business processes are all about alignment

Processes and functions – three key points

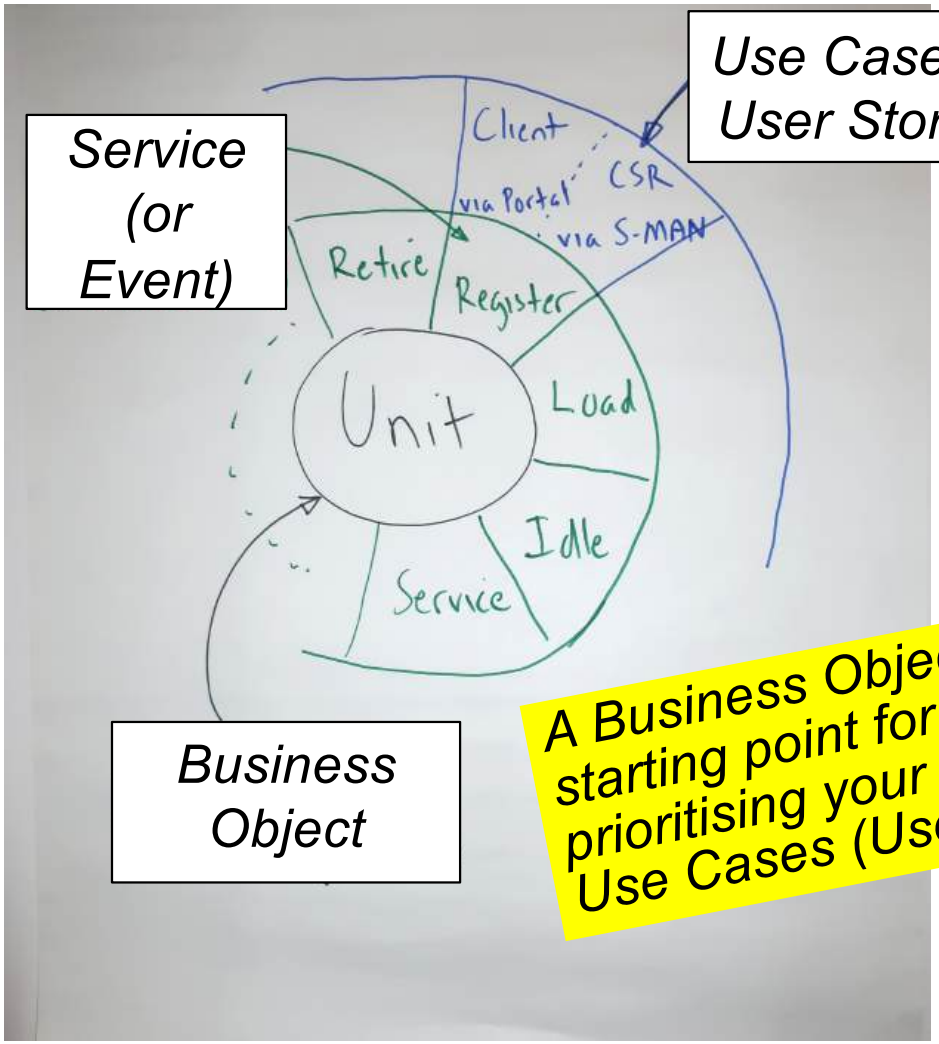


- The first step in managing processes is to *determine what they are* – they don't identify themselves
- Performance goals for the functions must *align with* (or be *balanced against*) the performance goals of the process
- Processes need an *owner / steward* to set direction, ensure alignment, and resolve conflict

It takes concerted effort – nothing happens by accident

Essentials of Use Cases & Services

Review – Business objects, services, and use cases



A Business Object Model is a great starting point for discovering and prioritising your Business Services and Use Cases (User Stories)

Review – Is there an alternative approach?

Simplistic methods at one extreme: can do as much harm as good

List-form requirements, typically a Business Requirements Document, (*context-free requirements)*

ID#	Business Feature	Requirement Type	Business Unit(s) Affected	Potential Application(s) Impacted
BRQ025	files that are available for the selected day.		Readiness	
OMSPI-BRQ026	System shall include all outage status in the Transmission Outage report.	Core	Operation Readiness	WebOMS
OMSPI-BRQ027	System shall display consistency in the format of output data in the Transmission Outage report when using pipe-delimited feature as follows: For the same row of output data, all data elements in the same position in any field must correspond to each other. Example of existing Transmission Outage report where there are two inconsistencies in the output data format: 1. Report shows one Outage ID, three Substations, and four Equipment Names. 2. First listed Substation does not correspond to the first listed Equipment Name. Outage ID: 3042750 | Substation: HUNTERS POINT PP P / MISSION X| LARKIN Y / POTRERO PP A (PGAE)| MISSION X | Equipment Name: A-Y 2| BNK-2| P-X 1| P-X 2	Core	Operation Readiness	WebOMS
OMSPI-BRQ028	System shall allow the format of the Transmission Outage report published periodically automatically to support the following formats: 1. PDF 2. HTML 3. MS Word	Core	Operation Readiness	WebOMS
OMSPI-	System shall allow admin user to configure the number of days in the Transmission	Core	Operation	WebOMS

The goal lies in the middle ground:

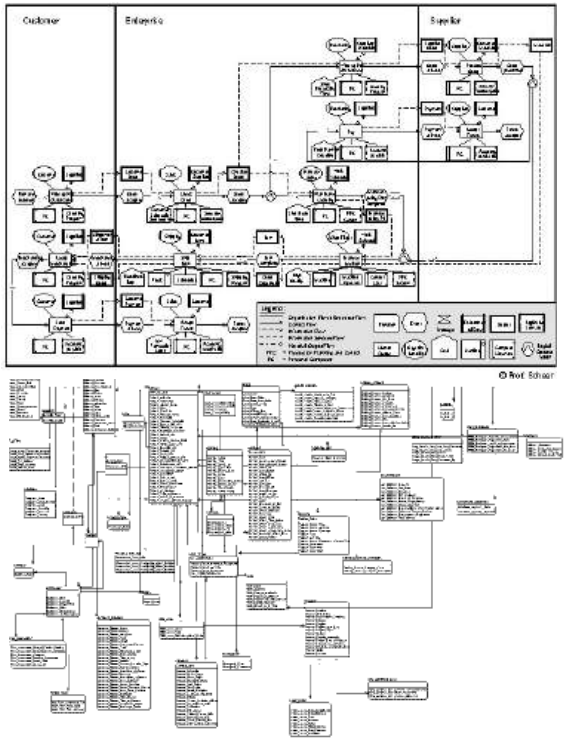
Client – *understandable, and therefore verifiable.*

Analyst – *doable, within their natural lifetime.*

Developer – *unambiguous, complete, actionable*

Overly complex methods at the other extreme: difficult for business people to verify

Thinly-disguised, implementation-level design methods – *not* useful for discovering stakeholder needs



Use cases to the rescue?

Use case - a description of a specific case in which an actor will use a system to complete a task or obtain a service

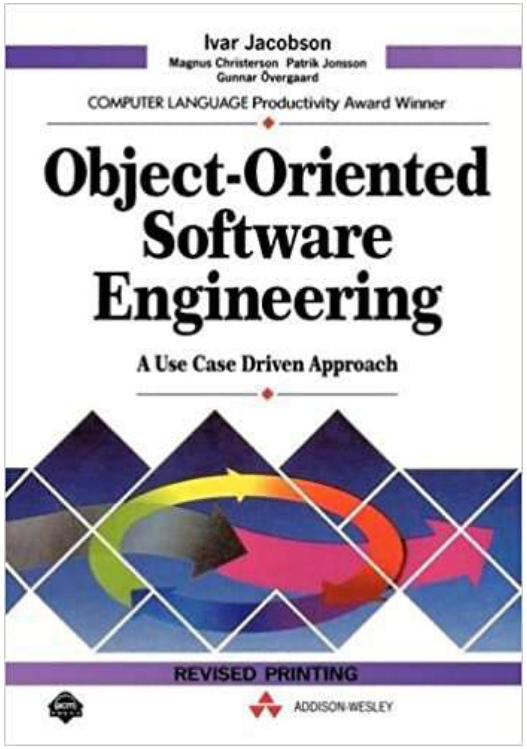
The idea – appealing in its simplicity

- Recognizable *tasks* provide *context.*

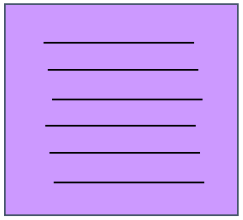
- *"Use cases are wonderful but confusing."* Alistair Cockburn
- *"A use case seems to be anything anyone wants it to be."* Charlie MacLachlan

The reality – plenty of grief and confusion

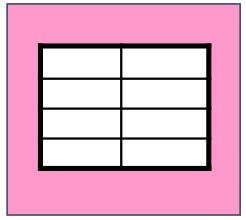
- Granularity, form, content, perspective, used for, …?
- How many use cases?
- Complete, self-contained methodology?
- Excessive complexity of some approaches



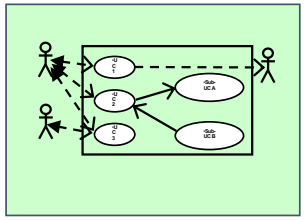
- Will the real Use Case please stand up? -



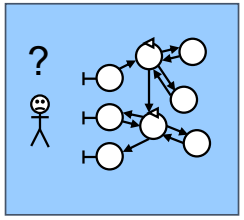
Meandering Narrative



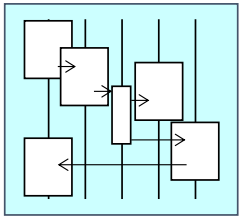
Tabular



UML Use Case



Ideal Object Model



Sequence Diagram

Traditional use cases can't do it all, part 1

Process

Business Process

- Workflow models are well understood
- Provides *context* – actors, tasks, sequence
- Helps to ensure we aren't automating bad processes or unnecessary tasks.

Application

Application requirements

The central question –
are *use cases* the best mechanism for discovering, documenting, and verifying *all* application requirements?

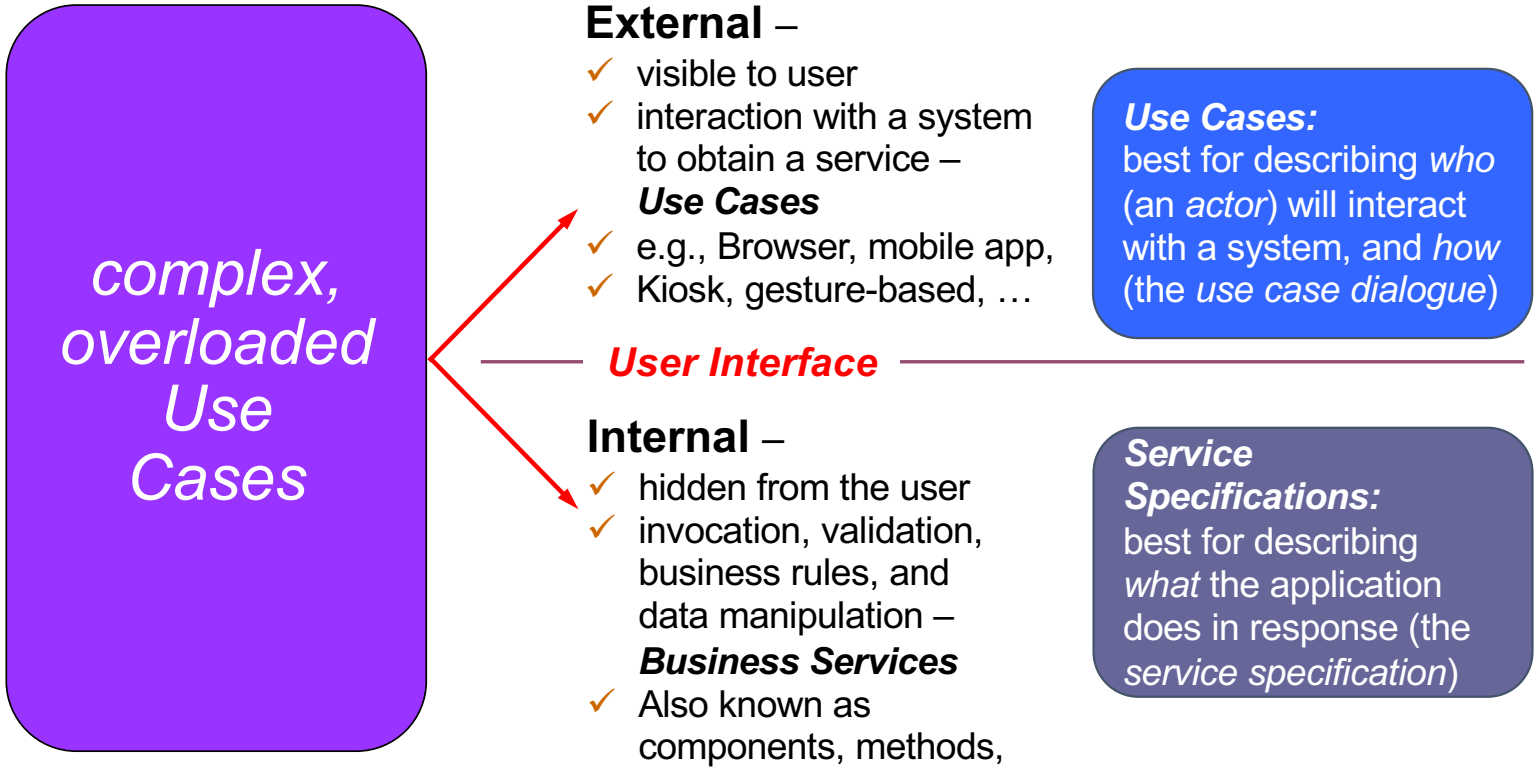
Data

Data Management

- Business object models *can* be well understood, with care
- Provides *platform* – vocabulary, structure, dependency
- Helps to ensure we're all using the same terminology, and employing the same basic concepts and rules

Traditional use cases can't do it all, part 2

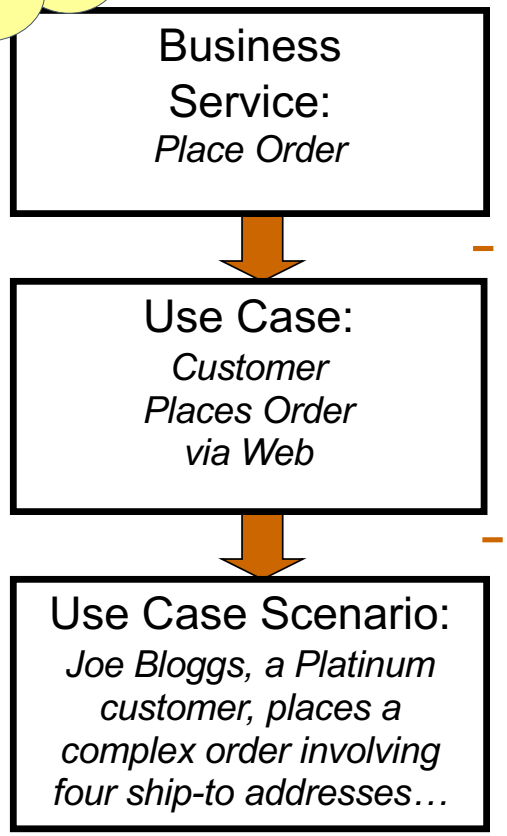
The central problem with many use case methods is a failure to recognise that applications have external and internal views –
"techniques that work for one don't work for another"



Services, Use Cases, Use Case Scenarios

Review, Check, Monitor, Track, Analyze, Enable, Handle, Process, Manage…

No mushy verbs!!!
"Noun is Verbed" (Order is Placed) must be an essential event.



- ✓ Abstract or "essential" - no reference to "who or how"
- ✓ Action verb + noun (+ noun)
- ✓ Helps us to focus on the *essence* of *what* must be accomplished to operate the business
- ✓ Often surprising to a business to see what it really does, stripped of all procedural overhead

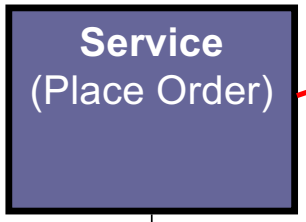
- ✓ Generalised (or "abstract")
- ✓ Actor + service (or goal) plus (usually) technology (browser, purpose-built kiosk, IVR, …)
- ✓ Helps us document different situations
- ✓ Same service can be accessed via multiple use cases
- ✓ Demonstrated in multiple UC scenarios

- ✓ Specific (or "concrete")
- ✓ A scenario comprising a "worked example" of one or more linked Use Cases
- ✓ Scenario – a story or "vignette" including named actors, specific data values, and predefined decision outcomes.
- ✓ Helps put UCs in context, so users can contribute / verify.

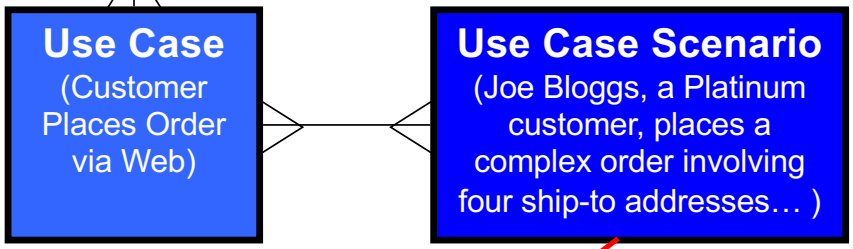
Relationships among Services, Use Cases, Use Case Scenarios

Key Point

We follow an "inside-out" approach – services first, then use cases



A single indivisible capability that must be carried out in order for the business to operate. It is the system's response (rules and data updates) to some trigger, and results in a business event



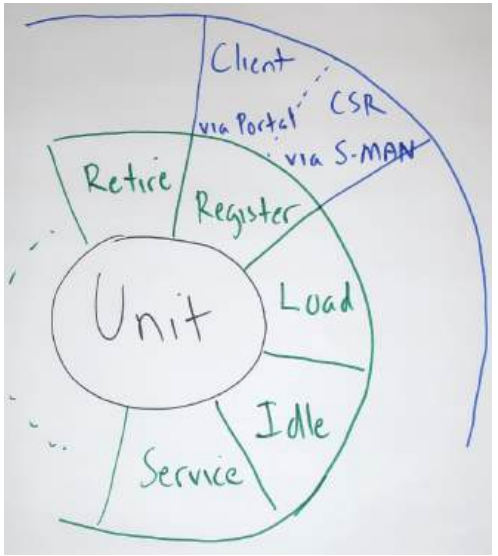
How a specific actor, with a specific technology, will interact with the system to obtain the service.

A "worked example" – a walkthrough of an actual session, encompassing one or more use cases, using named actors, predetermined data values, and predetermined decision outcomes. It is a single sequence of interactions with no branching or alternative flows.

Discussion – one Business Service, one or more Use Cases

One Service

	Who	What (the Service – verb + noun)	How
Multiple Use Cases	Client	Register Unit	via Portal
	Customer Service Rep (CSR)	Register Unit	via S-MAN (the ERP)
	Client	Register Unit	via Mobile App
	???	Register Unit	???



What is the value of documenting the Service only *once?* ("One Service available through multiple channels.")

- re-use of the asset, and therefore higher consistency
- better chance of getting it right – higher value from less effort
- if it's implemented as a single service, easier maintenance – it's in ONE place.

Why would we make a *single* Service available via *multiple* Use Cases?

- different actors need different "navigation and hand-holding," e.g., casual vs. expert users
- different technology platforms have different capabilities, e.g., mobile phone vs. touch-screen kiosk

Multiple discovery approaches

Choose *one* as your *primary approach,* use *others* to verify & extend – there is no *one best way*

We did this on Day 1

Good for completeness

Business object based

1. ID core business objects
2. List primary services (or events) for each
3. Describe service rules and actions, and optionally state diagrams and other service logic
4. Allocate services to actors, yielding Use Cases
5. Proceed with Use Case development

Good for big picture, context

Workflow based

1. Design new workflow
2. Isolate "system supported" steps, which are Use Cases for that actor
3. Proceed with Use Case development

Discovering Business Services and Use Cases

Good for relevance, buy-in

Actor based

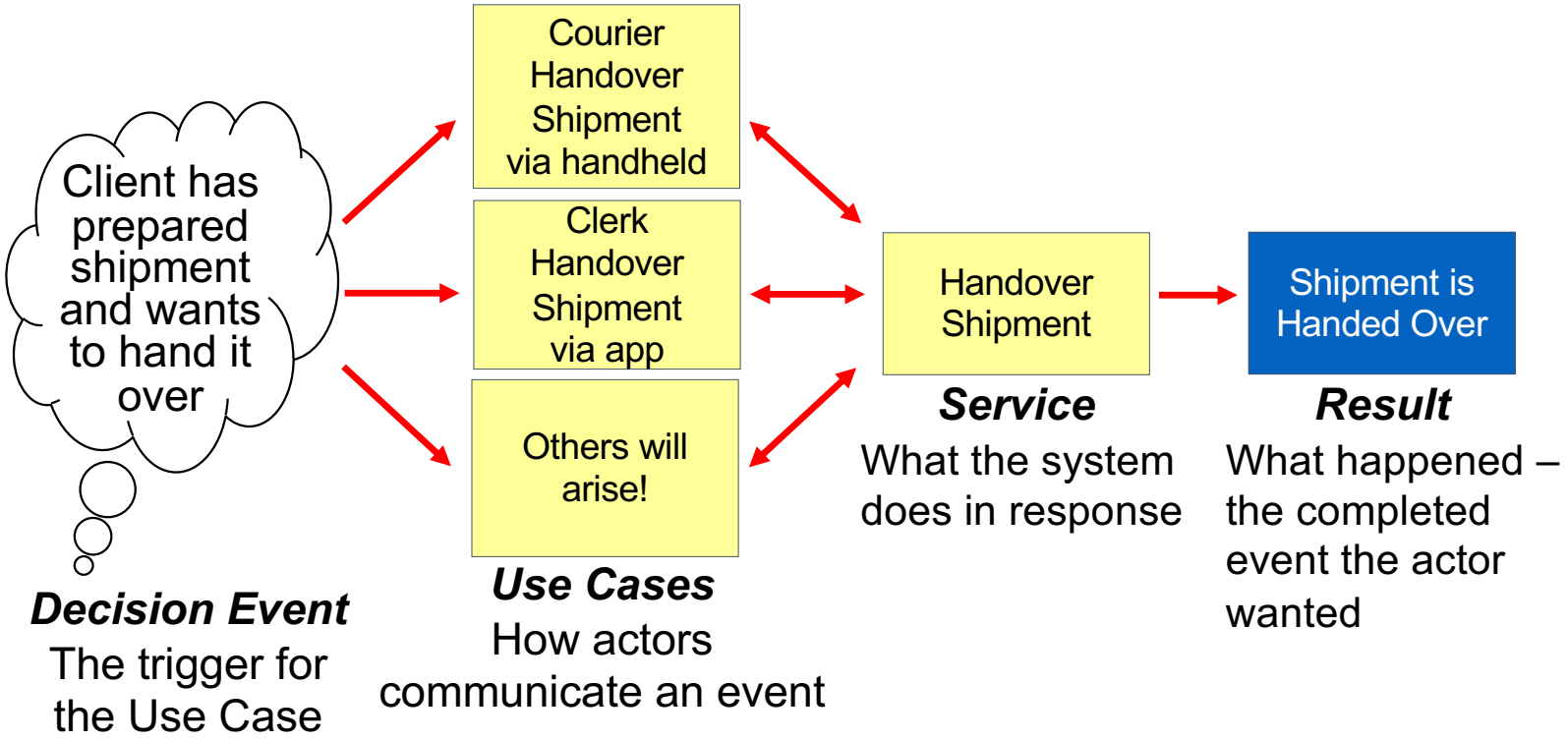
1. ID actor
2. ID services or goals (use cases) for each actor
3. Proceed with Use Case development

Good for finding outliers

Unstructured

1. What else happens that the system must respond to?

Use cases, services, events



You can look at a use case as a mechanism for:

- ✓An actor to notify the system that the actor *wants* the event to happen, e.g., Place Order
- ✓An actor to notify the system that an event *has* happened, e.g., Change Address and they want the system updated (after the fact)

Three kinds of events trigger services

Action Event or Decision-Based Event	*Temporal Event or Time-Based Event*	*Conditional Event or Data-Based Event*

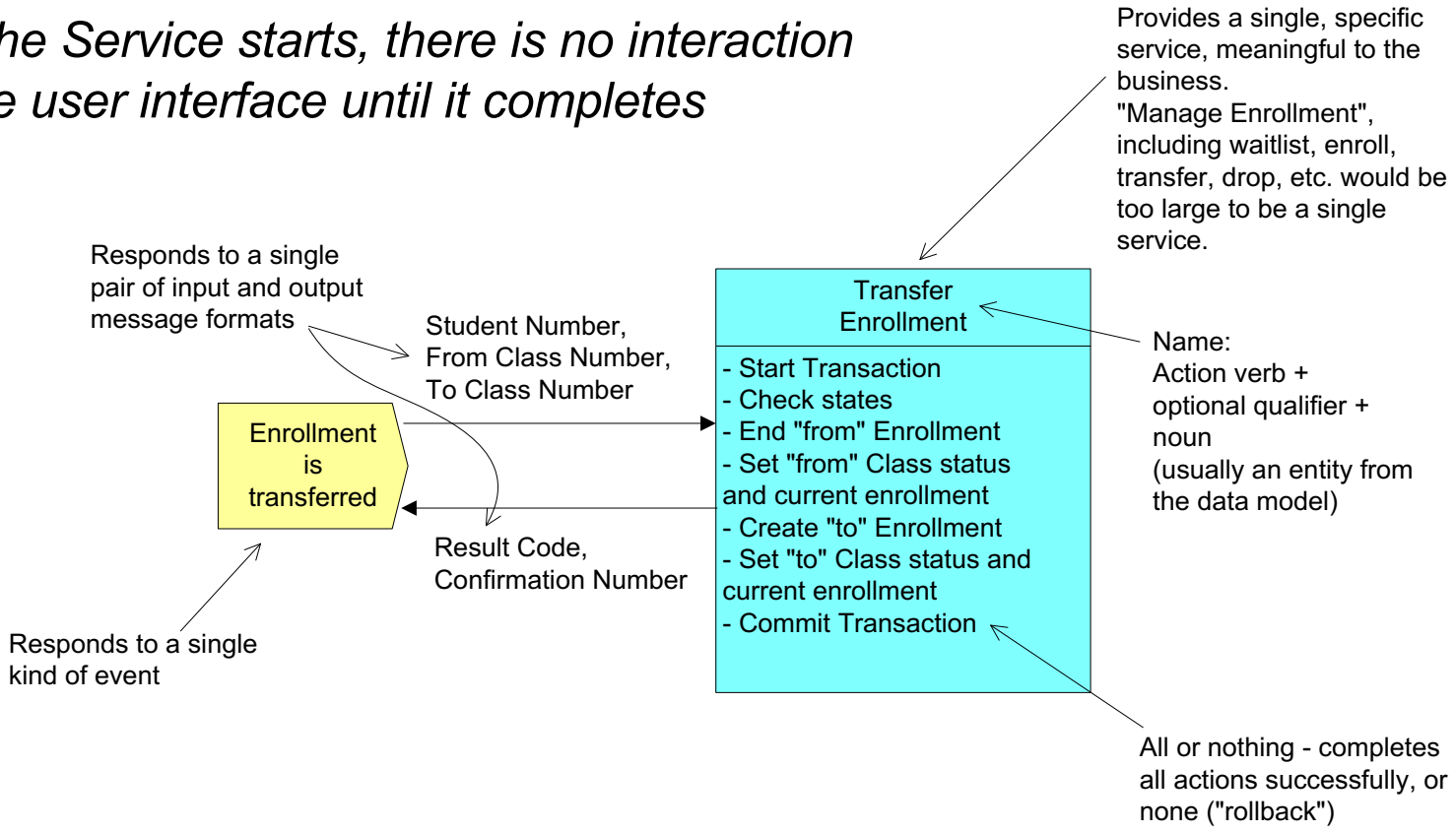
Raised by an actor deciding to do something
 	
E.g.,
• Decision to Place Order
• Decision to Raise Employee Salary
• Decision to Submit Complaint
	Raised by the system when a predetermined date/time is reached
	
	E.g.,
	• Time to Place Recurring Order: trigger Place Order
	• Time to Pay Employee
	• Time to Submit Financial Statements
		Raised by the system when a predetermined threshold is reached
		
		E.g.,
		• Inventory Reorder Level is Reached: trigger Place Order
		• Temp >0C & <40C:
		  • High temp threshold hit
		  • Low temp threshold hit

Always introduces new data to the system	*Does not* introduce new data to the system	*Might* introduce new data – the measurement and time
Needs a use case for a human actor to convey event to the system a.k.a. a Real Use Case	*Does not* need a use case, except to “set” the alarm a.k.a. a System Use Case	*Does not* need a use case, except to “set” the data threshold a.k.a. a System Use Case

Business Service guidelines

"What", not "who" or "how" – *completely independent of actor and user interface.*

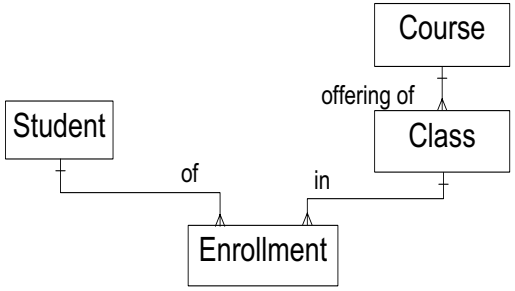
Once the Service starts, there is no interaction with the user interface until it completes



Initial (concept level) service specification

	Component	Description	Notes
1	Name	The Business Service name	• Typically "action verb + noun" or "action verb + noun + noun"
2	Result	A short (1 – 3 sentence) description of how the world (and therefore our records of it – files or databases) are changed by successful completion of the service	• Must use the language of the business object model and any other pre-defined artifacts (e.g., standard calculations like metrics) • Must make sense to both business and technical audiences
3	Action	5 +/- 2 (give or take) bullet points describing the key steps that comprise the service	• Again, uses the language of the business object model and any other pre-defined artifacts, and makes sense to both business and technical audiences • Focus is on "what, not how" and successful completion (not all the exceptions) • Will describe essential validation, and the core operations and data updates • Corresponds, in part, to "acceptance criteria" in a user story
4	Notes	Any additional requirements, assumptions, or questions that arise	• May include requirements (e.g., constraints or business rules) that will later be captured in the detailed service spec, or elsewhere, e.g. the use case or object model

Initial (concept level) service spec example



	Component	Description
1	Name	Complete Enrollment
2	Result	Enrolls a qualified Student in a single Class by creating an Enrollment record linked to Student and Class
3	Action	• Validate Student status and prerequisites • Confirm space in Class • Create Enrollment, link to Student and Class • Generate confirmation number • Revise remaining space in Class • Apply fee to Student Account
4	Questions & Assumptions	• Does this need to be able to create an Enrollment in "waitlist" status?

Now, review main actions with subject matter experts (SMEs) and ask:

- "Would you usually do *more* or *less* than this?" That is, is the service too *small* or too *big*?
- "Have we missed any important Actions?"