Business-Oriented Process & Data Modelling –
Proven Techniques and Synergies

A three-day workshop developed and presented specifically for Booking.com by
Adept Events and Clariteq Systems Consulting

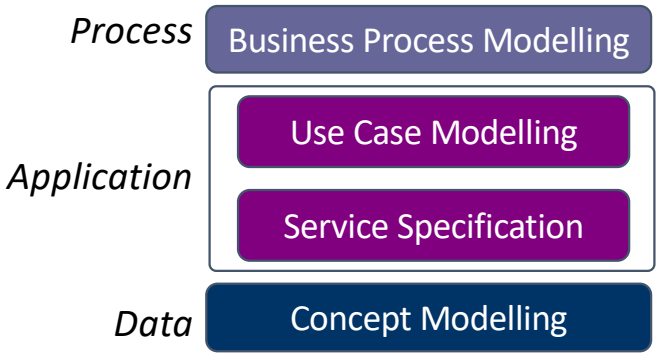
Alec Sharp
Consultant
Clariteq Systems Consulting Ltd.
West Vancouver, BC, Canada
asharp@clariteq.com
www.clariteq.com

Booking.com ///AdeptEvents CLARITEQ

Instructor / course developer background…

Alec Sharp, Clariteq Systems Consulting – *asharp@clariteq.com*

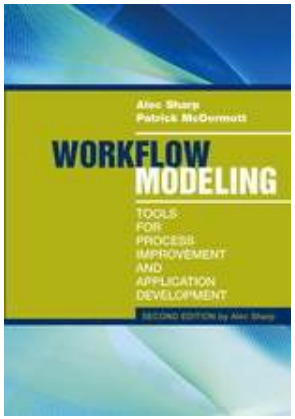
- 40+ years global experience as an independent consultant:
  - Business Process Modelling & Business Process Change – discover, scope, analyse, and design/redesign processes
  - Application Requirements Specification
  - Data Modelling and Management My roots!
  - Facilitation & Organisational Change
  - Project Recovery



- Awarded DAMA's global Professional Achievement Award for contributions to "human-friendly" data modelling

- Author of "Workflow Modeling"
  - best-selling book on process modelling & improvement
  - second edition – a complete re-write

Check out the nice reviews on Amazon - http://amzn.to/dHun1o



Clariteq – small, husband & wife company, global clients

ABB (ASEA Brown Boveri)
Aflac
American Honda
AMP (Australia Mutual Provident)
BackOffice Associates
Bank of Finland
Bellrock
Booking.com
Brisbane City Council (Australia)
Canadian Natural Resources Ltd.
City of Seattle
Civica UK
Clearwater Paper
Corvias
Dell
DHL Express
Dutch National Bank
Elisa
Ericsson
Essity
Eurojust (European Justice Comm.)
European Central Bank
Fortum
Gofore
Helse Vest - Norway
HM Land Registry - UK
Home Depot
Idaho Transportation Dept.
Intel
ISO New England
ING Bank
JP Morgan
Kal Tire
KONE
LGM Financial Services
Liberty Mutual
Livestock Improvement Corp.
MacDonald Dettwiler
Manitoba Public Insurance
Marathon Pipe Line
Microsoft
Ministry of Defence - UK
Ministry of Defence - NL
Ministry of the Interior - Slovakia
MTS Allstream
Nexen
Novo Nordisk
Nusenda Credit Union
OP Bank
Partner Reinsurance
Ritchie Brothers
Phillip Morris
Roche Diagnostics/Pharmaceuticals
Salt River Project
Saudi Aramco
Serco
Shell
Sparta Consulting
State Street Bank
SunGard
SVB (NL)
Synechron
Sysdoc
Talent Base
Teck
The MUSIC Group
The Seattle Times
UK Government
University Med Ctr Groningen
YIT(FI)
Washington Gas & Light

– Higher Education –
Carnegie Mellon University
Cornell University
Douglas College
Gonzaga University
Humboldt State University
The Jackson Laboratory
The Ohio State University
Portland State University
Salt Lake Community College
Southern NH University
University of Arkansas
University of British Columbia
University of the Fraser Valley
University of Maryland
University of Utah
University of Washington
Utah Valley University

Background for this course

3 days
(2-day version developed for Booking.com)

Business-Oriented Data Modelling Masterclass

3 days

Working With Business Processes Masterclass

Business-Oriented Process & Data Modelling

3 days

Learn techniques for modelling more complex situations.
Two important notes:

1. Learn proven techniques for process and data modelling, and specific methods for integrating them.
2. The focus is on simplicity and engaging businesspeople in directly in modelling, and improving the organisations processes, information, and systems.

Overview and logistics

Course Topics

1. Requirements Definition
   - Goals, Issues, and the Return of Modelling
   - Case Study - Integrating the Techniques
2. Business Process Fundamentals
   - Five Things You Need to Know
   - Discovering, Scoping, & Assessing Your Processes
3. Concept Modelling Fundamentals
   - E, R, A - A Concept Model's Essential Components
   - Drawing Your Model for Maximum Understanding
4. Business Process Workflow Modelling & Design
   - Five Core Guidelines for Great Swimlane Diagrams
   - Facilitating a Process Mapping Session
   - Assessment of the As-Is and Transition to the To-Be
5. The Process-Data Connection
   - The Natural Synergy between Process & Data Models
   - Process-Data Synergies in Modelling, Analysis, & SW

Schedule (CET)

- 09:00 start
- 09:00 - 10:30 *class*
- 10:30 - 10:45 break
- 10:45 - 12:30 *class*
- 12:30 - 13:30 lunch
- 13:30 - 15:00 *class*
- 15:00 - 15:15 break
- 15:15 - 17:00 *class*
- 17:00 end

Finally…***you*** –

- Name (how should I address you?)
- Brief description of your work
- Is there a topic you are especially interested in?
- *Please try to keep your introduction to one minute or less*

"Business Analysis" gets criticised because of the extremes

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 | Substation | Equipment Name 3042750 | HUNTERS POINT PP P / MISSION X| LARKIN Y / POTRERO PP A (PGAE)| MISSION X | 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:

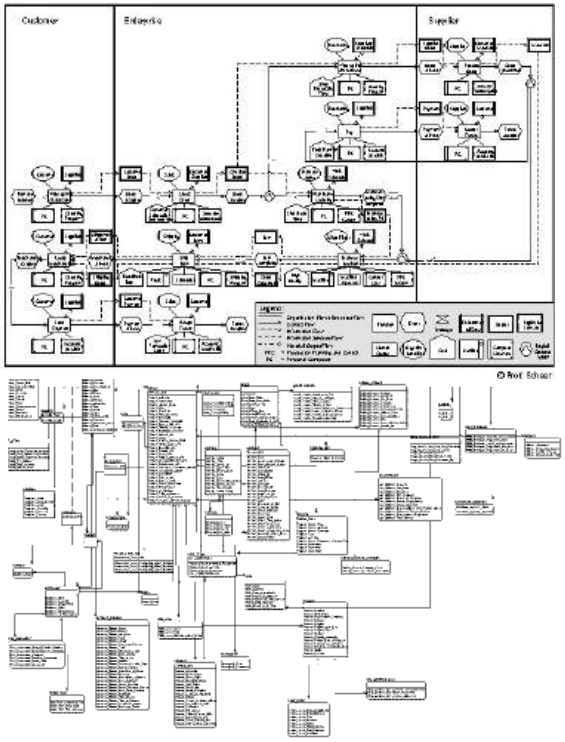
Client – *understandable, and therefore verifiable.*

Analyst – *doable, within Agile timeframes.*

Developer – *unambiguous, complete, actionable*

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

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



Discussion – the problems with list-based requirements

Simplistic methods at one extreme:

An actual example, one in a list of 451 individual requirements for the "Provide Scientific Evidence" process at a national forensic science laboratory:

#49 -

> *The system shall provide a visual mechanism through which to view or amend the sequencing of items for a previously selected case or allocations thereof.*

WHAAAT???!!!

List-based approaches to business analysis quickly break down – no way to ensure *completeness, accuracy, consistency, …*

So… what's wrong with this as a requirement? What does it NOT tell us?

What are they really trying to say?

Who? Senior Scientist

What? Schedule a Test (an Allocation) on a Sample from an Item

When? At Item Submission

How? By viewing upcoming workload

Why? To provide a completion date to the Customer (the Police)

Essentially, a Use Case or *User Story*:

As a Senior Scientist, I need the ability to view upcoming workload and schedule a Test on an Item, so I can provide a completion date to the Customer.

We will also use

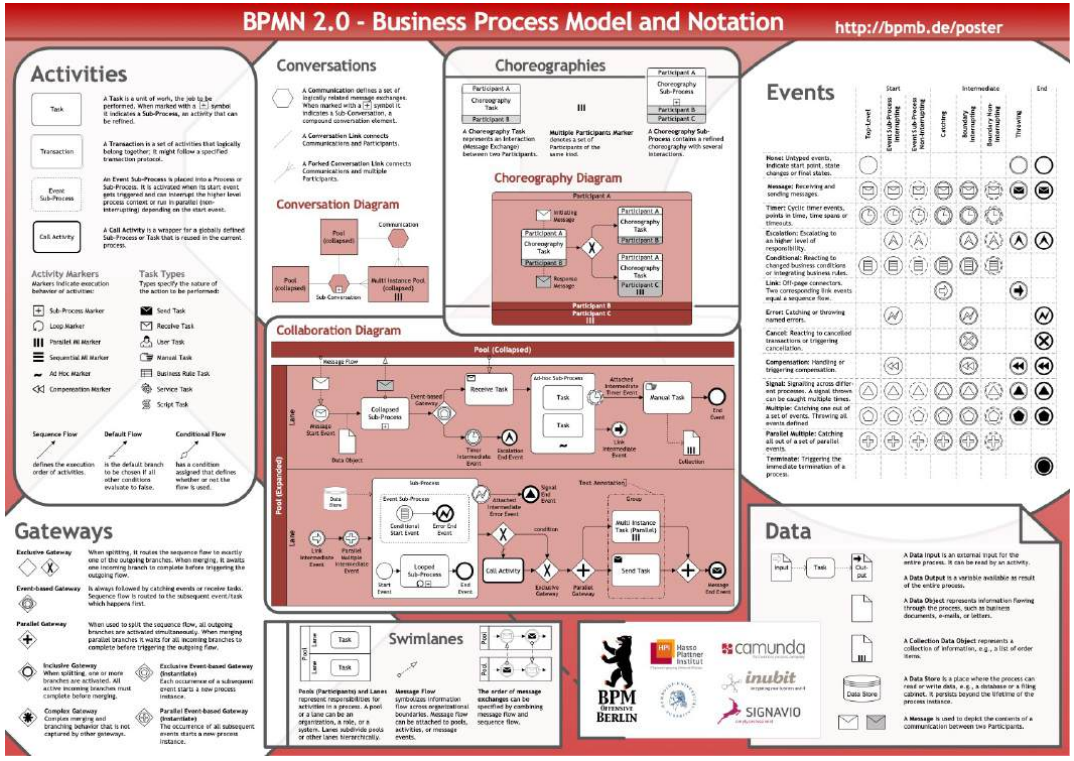
- *Business Process Models* to show where this fits in the end-to-end process
- *Concept Models* to show the required information

Complicated methods at the other extreme

"Can we use UML for Business Analysis?" As the late Michael Hammer said:
“You could, but it will be like eating rice with a steak knife –
messy, and someone’s going to get hurt.”

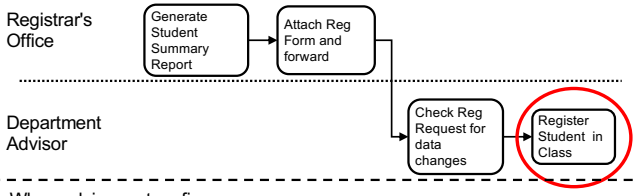
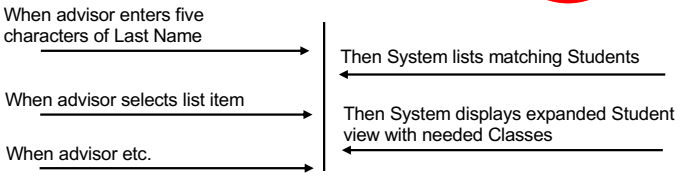
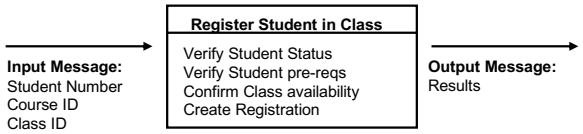
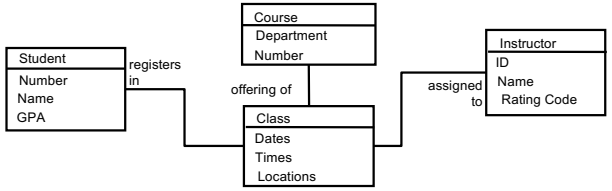
From the original UML specification:
“The Unified Modeling Language (UML) is a graphical language for visualizing, specifying, constructing, and documenting the *artifacts of a software-intensive system*.”

Same story for full BPMN (Business Process Model & Notation) – a platform-independent *visual programming language* for specifying automated workflows.



Data Models in an integrated, model-based framework

The Clariteq Framework for Business Analysis

Framework Layer	Technique sample	What it covers
Business Goals & 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
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
Presentation Layer (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 Layer (rules & logic)		✓ ***Service Specification*** - describes a service – a package of rules and logic – that is triggered to complete or respond to a business event
Data Layer (data & storage)		✓ ***Concept Model*** - depicts the things and the facts about things the organisation needs to record; the things (the entities) are what processes and solutions act on.

This is not a sequence!

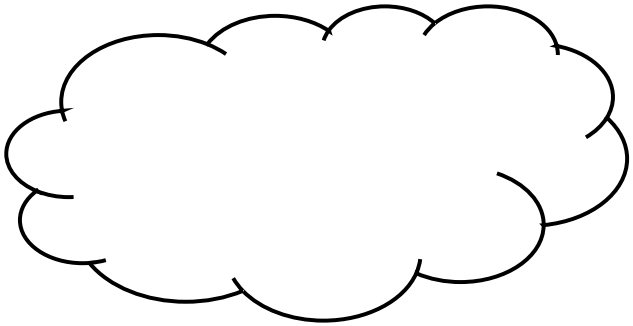
Business Process: gives great *context* for *Business Analysis*

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

Concept Model / Data Model: a great *platform* for *Business Analysis*

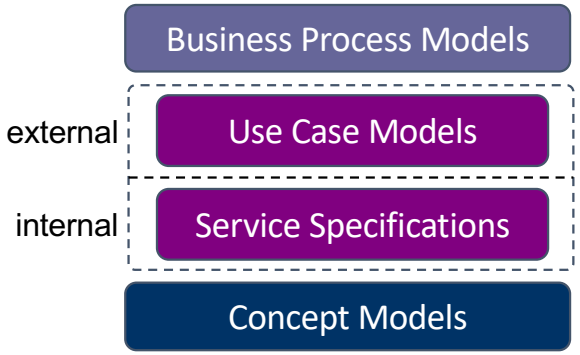
These techniques supports the Business Analyst's role – current to future state

Current state



Iteration

Future state



The current *(as-is)* state may be incompletely understood, with multiple perspectives – some *as-is* modelling will clarify it.

The future (*to-be*) state, described with a rigorous set of interrelated models – *easier* than you might think

- A "best practice" solution – *"integrated business modelling techniques"* which build graphic and narrative models highlighting different aspects of the *as-is* and *to-be* states
- Key ideas – "business-friendly," "suitable for mere mortals," "progressive detail," "high context," and "do-able within your natural lifetime"

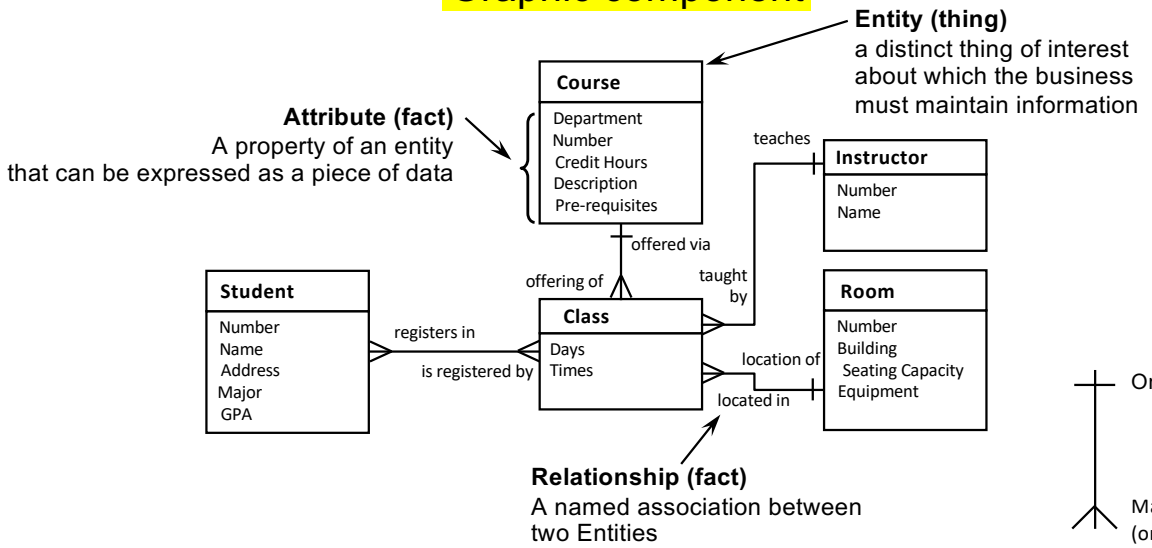
We'll start *(surprisingly!)* with the Concept Model / Conceptual Data Model

What is a Concept Model / Business Object Model / Domain Model…?

- A *description of a business* in terms of
  - **things** it needs to maintain records of – *Entities*
  - **facts about those things** – *Relationships & Attributes*
  - **policies & rules** *governing those things and facts*
- Models a view of the **real world**, not a technical design (therefore, stable and flexible)
- Can be comprehended by mere mortals (at least initially)
- Graham Witt – "A narrative supported by a graphic"

"Things" first, data later!

Graphic component



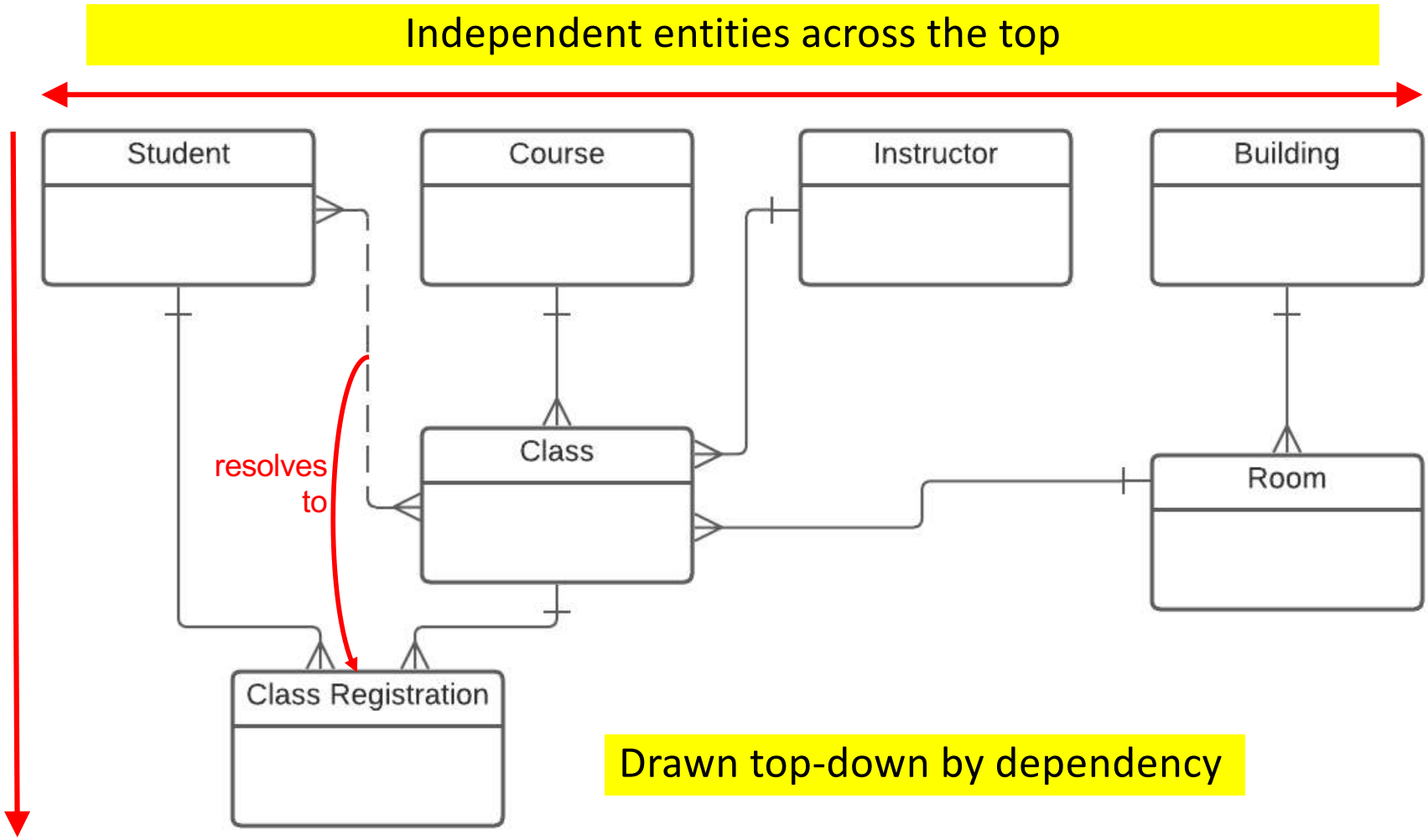
Narrative component

Student definition:
A Student is any person who has been admitted to the University, has accepted, and has registered in a course within a designated time. Faculty and staff members may also be Students.

Plus "Assertions" (policies & rules)
- Each Course is offered through one or more Classes
  Each Class is an offering of a single, specific Course
- Each Instructor teaches one or more Classes
- Each Class is taught by one Instructor
  (which may or may not be true…)

Many rules can't be shown on the diagram…
- A Student can not register in two Classes of the same Course in the same Academic Term

A better looking version of the model on the previous slide



Case study – Concept Model, Services, Use Cases, Business Processes

Client –

- Regulatory agency ensuring the safe design, installation, and use of technical equipment
- Natural gas systems, electrical systems, boilers and pressure vessels, elevating devices, & many more

Goal –

- Shift from an inspection-based model (~800 inspectors!) to client-managed safety programs
- Clients will apply for a *Client Safety Management Program Authorisation (CSMP Authorisation)* - must show effective processes and accurate record-keeping
- Clients will pay a fee for managing *their own safety programs!* Still beneficial!

Case study – Concept Model, Services, Use Cases

- Business Development chooses Pilot Program –
  boilers and pressure vessels in Oil & Gas fields



- Current systems won't support CSMP, time-consuming and expensive to change them –
  IT and Finance suggest 18 – 24 months of work
- BD is unimpressed by IT and Finance objections ("You're being mindlessly obstructionist!")
  and proposes work-around procedure. *Guess which tool they intend to use?*
- I'm hired to identify end-to-end implications –
  "Design a process and determine IT requirements that will allow this procedure to work."
- *Concept Modelling was a critical tool in understanding the underlying policies,*
  and developing the process & requirements

Building your initial Concept Model, step-by-step

Identify and define "Things"

1. Collect terms
- 1:1 interviews
- survey (e.g., email)
- group brainstorm
- analyse documents

2. Isolate "things"

Ask *Is this…*
- a thing?
- a fact about a thing?
- or "other stuff?"

3. Identify synonyms
- select a term to use
- as general as possible
- just for this initiative, not the entire enterprise

4. Define each thing
- "good enough for now"
- first, identify "anomalies, sources of confusion, and valid differences of opinion"
- select which to include

Develop initial Concept Model

5. Organise things
- independent things across the top
- then laid out top-down by dependency

6. Draw relationships
- show dependency
- parent-child drawn bottom-to-top
- otherwise, side-to-side

7. Name relationships
- in both directions
- active verb-based!
- not mushy – *has*
- not meaningless – *related to*

8. Add cardinality
- use words first
- 1:1 is probably wrong
- 1:M (one to many)
- M:M (many to many)

Refine Concept Model

9. State assertions
- forcefully, for each relationship
- challenge the assertions!
- restate the assertion & why it changed, if it did

10. Redraw the model
- shows revised assertions
- e.g., 1:M becoming M:M
- e.g., dependent things becoming independent

11. Collect attributes
- a few for each thing
- not *all* attributes
- don't worry about normalisation

12. Move to identifying:
1. events / services
2. use cases / user stories

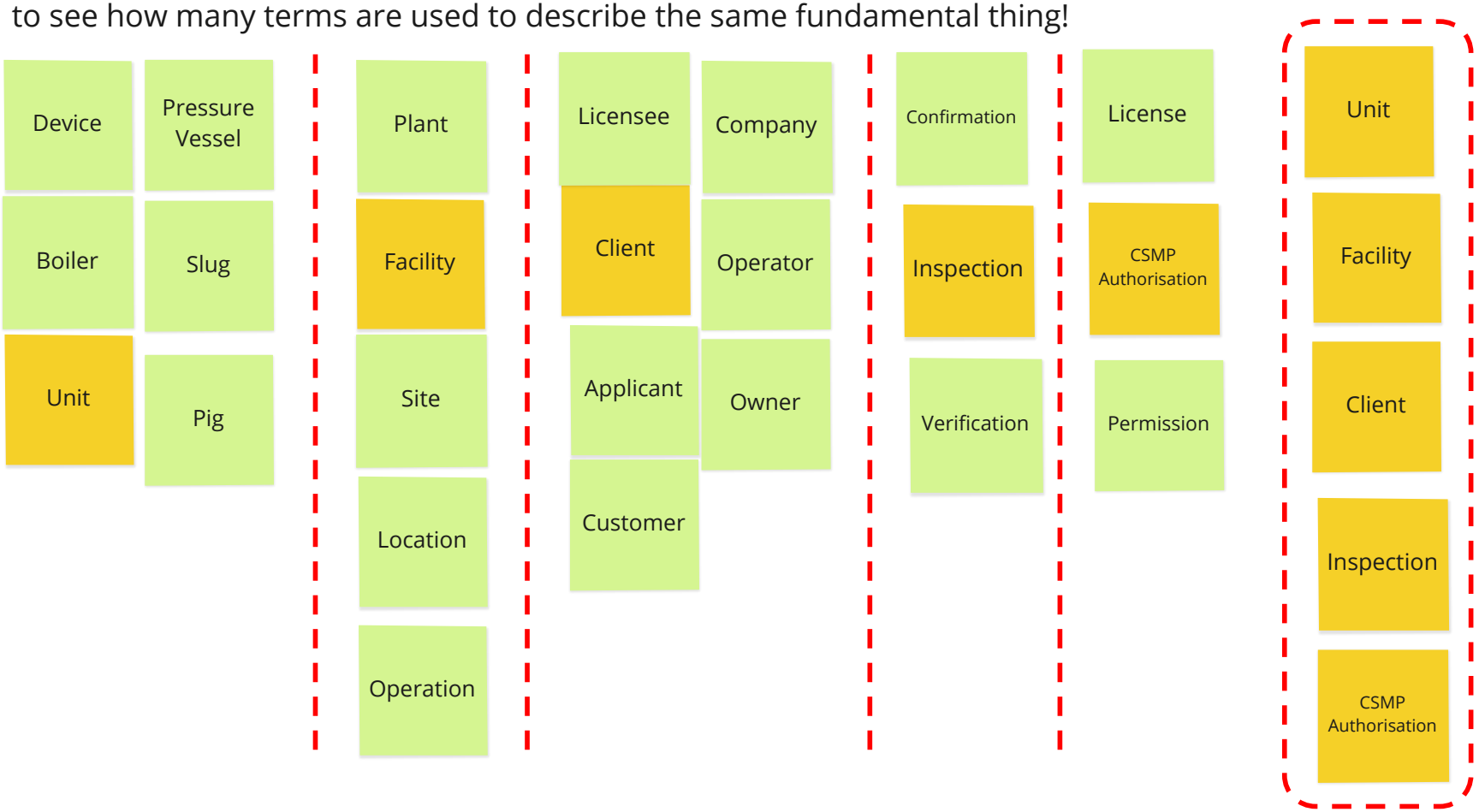
Always start with terminology (the "things")

From one-on-one interviews with 8-10 key stakeholders we gathered ~200 terms related to CSMP (Client Safety Management Program) – "anything that went by a name."
Here are 24 that met the criteria to be a "thing"– the candidate *Entities.*

Identify synonyms and select one term.
How do these relate to one another?
What do you need to know about each?

Review of a Miro example – Terminology Analysis

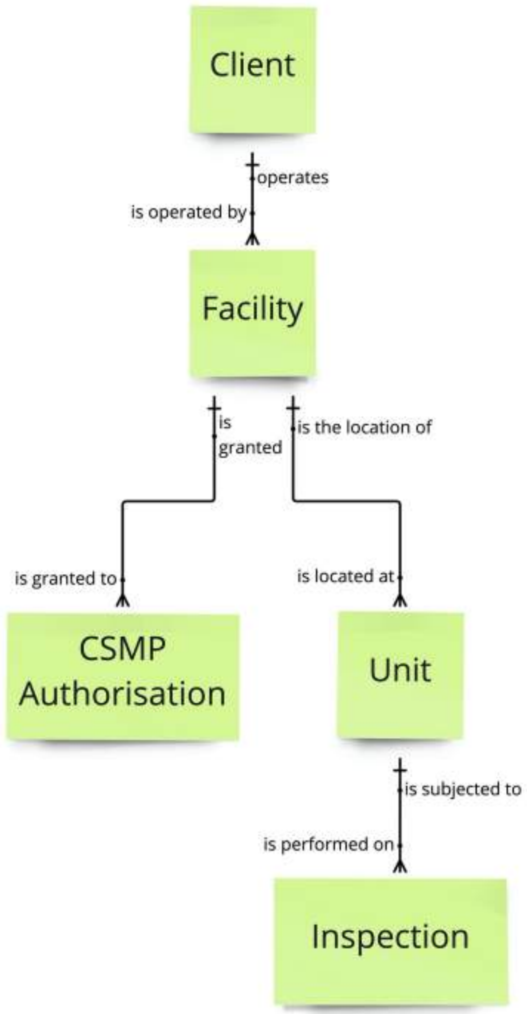
Terminology analysis (continued):
Let's arrange these terms into columns of synonyms. It's always a surprise for the business to see how many terms are used to describe the same fundamental thing!



BOPDM: Business-Oriented Process & Data Modelling

Concept Model Version 1 – not perfect, but a good start

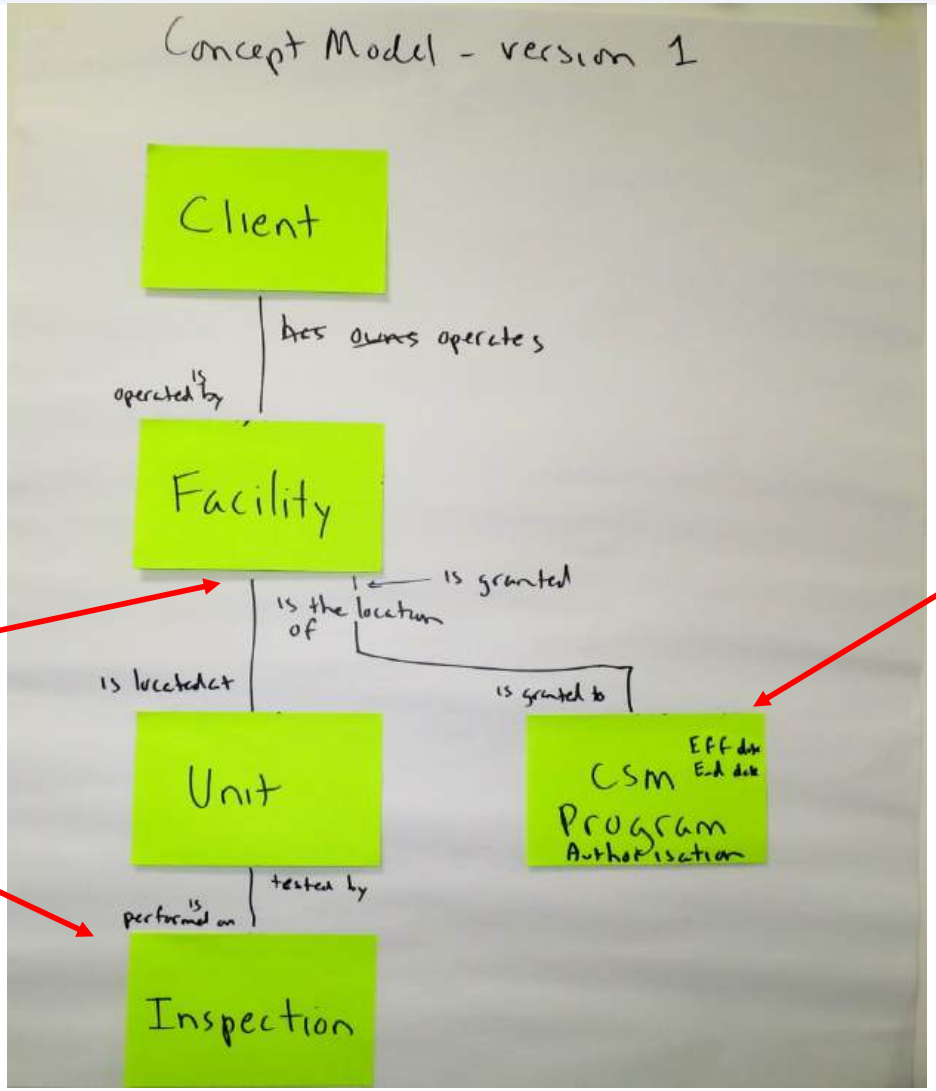
1. We arranged the entities / business objects by dependency
2. Then we drew relationship lines
3. Then we added a relationship name in each direction
4. Only then did we state (in words) the cardinality (1:1, 1:M, M:M) and then update the diagram with hash marks (┼) and crowsfeet (⋔)



Definition -
A CSMP Authorisation is a permission (or license) to operate a self-managed safety program (a Client Safety Management Program) at a specific Facility, for a specified time period, usually 1, 2, or 5 years.

The CSMP Authorisation is "all or nothing" - it covers ALL the Units at a Facility.

Just boxes and lines, but raises important questions



Are Units permanently part of one Facility?

What do we Inspect?

What do we issue the Authorisation to?

Concept Model Version 1 – state Assertions and challenge them

Now, state the relationships ***emphatically*** as Assertions. ***Each*** Client operates ***one or more*** Facilities! Then, ***challenge*** them!
Again, don't worry yet about ***optionality*** – whether the relationship ***must be*** or ***may be*** be present.
We only care now about the ***maximum*** – each ObjectA is related to a ***maximum*** of ***one*** or ***one or more (or many)*** ObjectB.

Assertion:
Each Client operates ___

Assertion:
Each Facility is operated by ___

Assertion:
Each Facility is granted ___

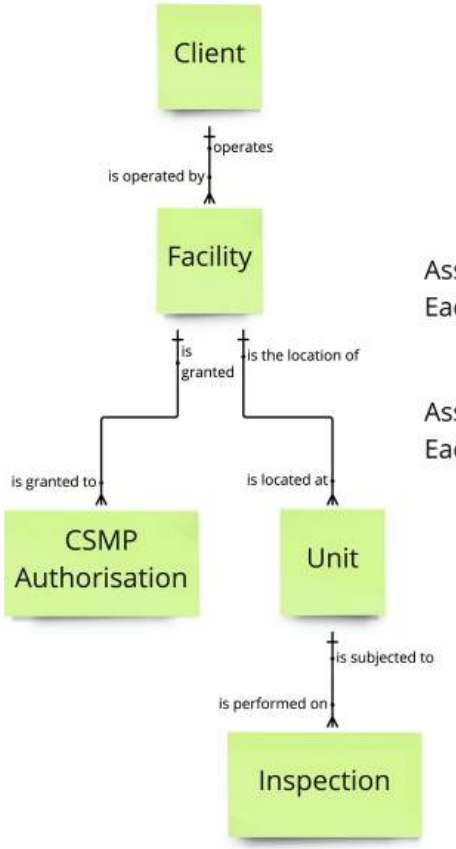
Assertion:
Each Facility is the location of ___

Assertion:
Each Unit is located at ___

Assertion:
Each CSMP Authorisation is granted to ___

Assertion:
Each Unit is subjected to ___

Assertion:
Each Inspection is performed on ___



Concept Model Version 1 – revised Assertions from challenges

Now, state the relationships ***emphatically*** as Assertions. ***Each*** Client operates ***one or more*** Facilities! Then, ***challenge*** them!
Again, don't worry yet about ***optionality*** – whether the relationship ***must be*** or ***may be*** be present.
We only care now about the ***maximum*** – each ObjectA is related to a ***maximum*** of ***one*** or ***one or more (or many)*** ObjectB.

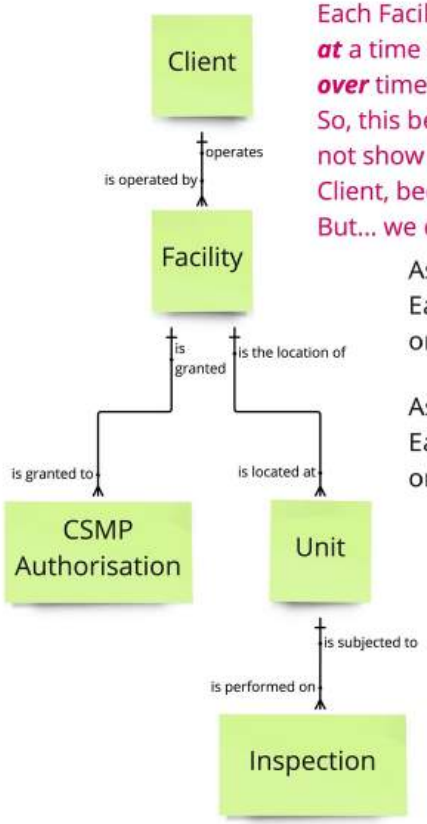
Assertion:
Each Client operates
one or more Facilities

Assertion:
Each Facility is operated by
one Client

Assertion:
Each Facility is granted
one or more CSMP Authorisations

One CSMP Authorisation ***at*** a time,
but one or more ***over*** time

Assertion:
Each CSMP Authorisation is granted to
one Facility



Each Facility is operated by one or more Clients
at a time (Joint Ventures) and
over time (changes in Ownership or Lease.)
So, this becomes a M:M relationship, and we should not show a Facility as being dependent on a single Client, because a Facility is an independent thing.
But... we don't always get our way!

Assertion:
Each Facility is the location of
one or more Units

YES, but one or more Facilities ***over*** time, because Units can move between Facilities. So, this becomes a M:M relationship, and we cannot show a Unit as being dependent on a single Facility, because a Unit is an independent thing

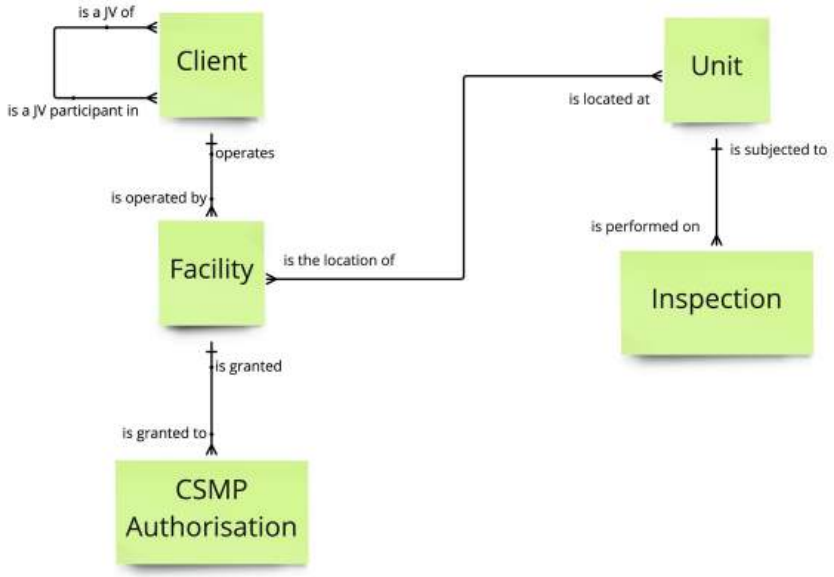
Assertion:
Each Unit is located at
one Facility

Assertion:
Each Unit is subjected to
one or more Inspections

Assertion:
Each Inspection is performed on
one Unit

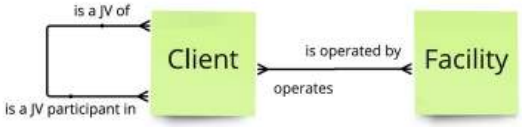
Concept Model Version 2 – revised from challenging Assertions

Now we will re-draw the initial Concept Model based on changes that came from challenging the Assertions in Ver. 1.

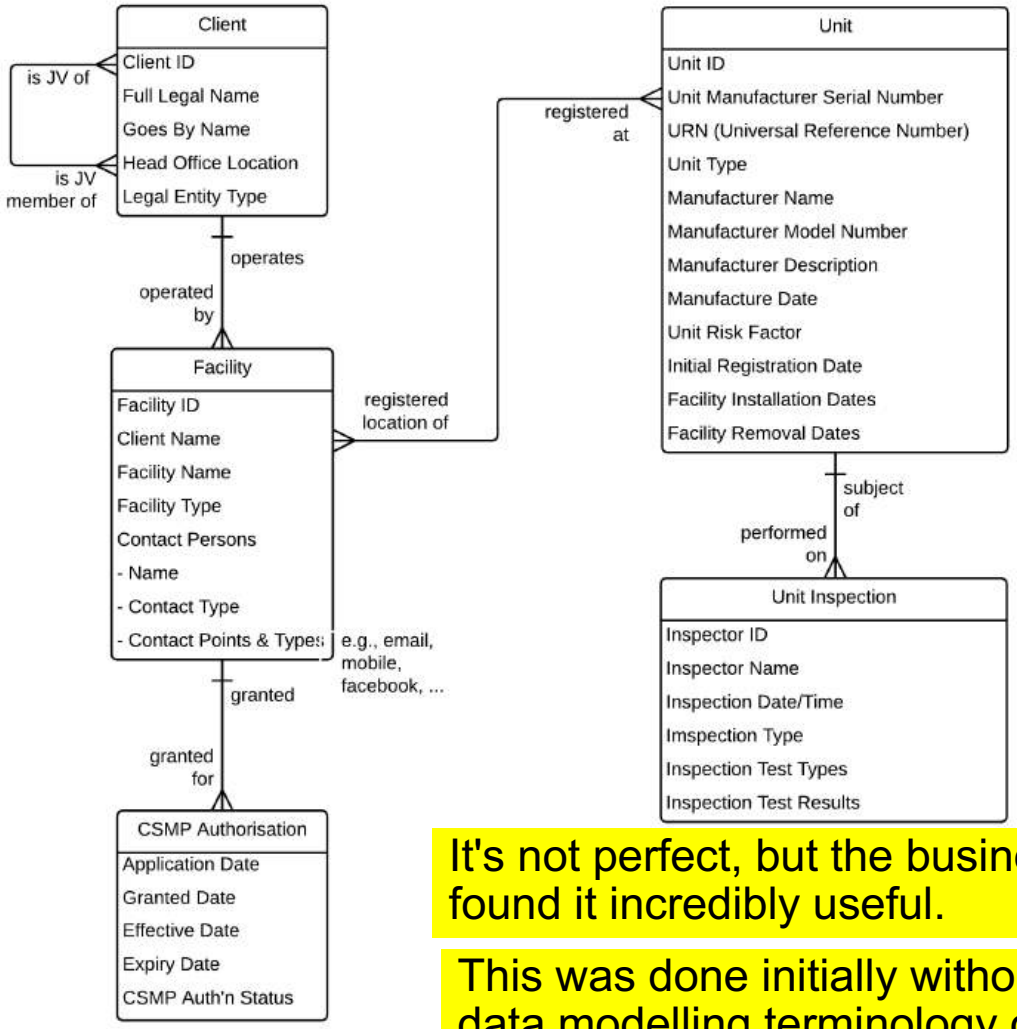


Note:
You don't always get what you *want* or what you think is the *right* thing in Concept Modelling. In this case the client (the Regulator) said they always wanted a Facility to be operated by ONE AND ONLY ONE Client.
If a Facility was operated by multiple Clients, they would require the Clients to form a new Joint Venture Client. This was to ensure that if there were legal difficulties, there was only ONE Client to go after.
Or, as they put it, "one throat to choke."

Later in the project, they realised they needed a history of the Clients that had operated a Facility, so the Client-Facility relationship became Many-to-Many, and Facility was modelled (correctly) as an independent Entity, as shown here:



"What do you need to know about the things in the Concept Model?"



Sketching this out was *fast, and* raised many questions that had not occurred to the client…

- Is there one CSMP per Client, per Facility, or some other basis?
- Do Units frequently relocate, or even turn up at another Client?
- What is inspected – the Facility or the Unit?
- Does the CSMP cover all or some Units at a Facility?
- …and MANY more…

It's not perfect, but the businesspeople found it incredibly useful.

This was done initially without any data modelling terminology or symbols!

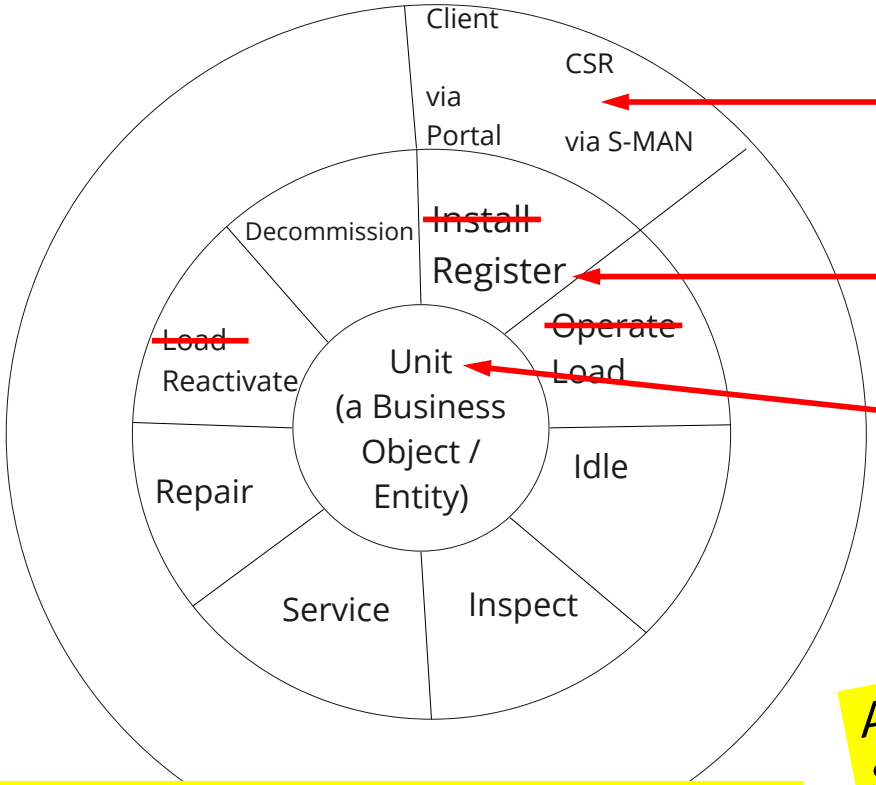
Model took ~90 minutes

Identify Services (Events) then Use Cases / User Stories

Finally, we'll identify the Services (verb - noun pairs) we need, and the Use Cases / User Stories by which the Services will be accessed

What events happen to a Unit - what are the needed services? (Verb - Noun)

- ...
- ...
- ...
-



Who needs access to each Service, and How?

Use Case	*Use Case or User Story - add Who and How*
Service Specification (Events)	*Service (or Event) - add a Verb to the Noun*
Concept Model	*Entity or simply a "thing" - a core Noun*

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

Supports *Service-Oriented Business Analysis*

Note – "User Story" and "Use Case" are not so different

Different format and detail, but the same basic concept.
Initially, at the Scope level, they're much the same:

User Story (who – what – *why*):
"As a Client, I need the ability to Register Unit(s,)
so I can maintain compliance with my CSMP Authorisation"

Use Case: (who – what – *how*):
"Client Register Unit via Portal"

When we add detail at the Concept level, they become identical:

- User Story / Use Case abstract
- Main success sequence – dialogue in "when-then" format
- Alternate sequences – variations, exceptions, errors

Develop high-level use cases and services

Service: *Register Unit*

- Check for presence of properly formatted UR Number
- Determine if Unit UR Number is previously known
- If known, has it (a) moved (b) changed ownership (c) …?

Use Case: *CSR Registers Unit via S-MAN*

- CSR will select "spreadsheet" of all Units covered by CSMP app
- S-MAN will highlight all that can proceed immediately
- For each category of Units requiring intervention…

Note:

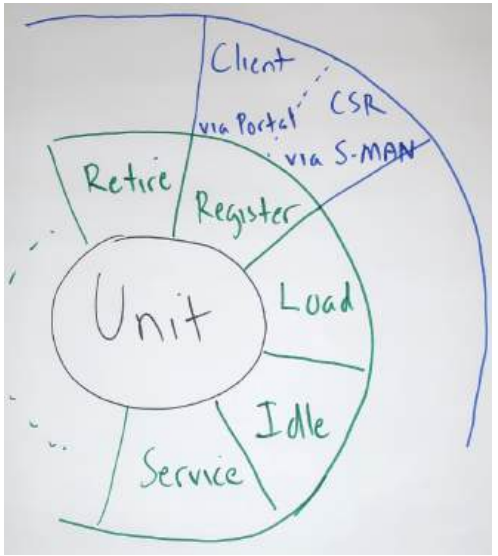
Services and Use Cases at the "upper conceptual" level to provide vendor with key elements of requirements and avoid the usual bulleted list requirements document.

Discussion – one Business Service, one or more Use Cases

One Service

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

Multiple Use Cases



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

Summary – what can an analyst do with a Concept Model?

First, clarify language. (A platform)

Second, establish policies and rules.

And then, identify events and services, e.g.,
A **Unit** is…

- Registered (requiring the service “Register Unit”)
- Loaded (requiring the service “Load Unit”)
- Idled (requiring the service “Idle Unit”)
- Reactivated (requiring…)
- Repaired
- Inspected
- Relocated
- Retired
- …

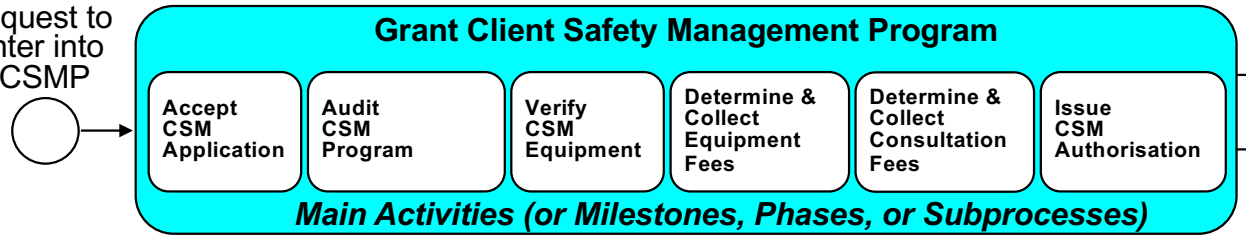
These are the *essential* capabilities. In Business Analysis "essential" means **what** with no reference to **who** or **how**

Something I always do when evaluating/selecting COTS S/W

We did the same for Client, Facility, CSM Program, …

Clarify scope of the new process and identify participants

Trigger:
Client submits request to enter into a CSMP



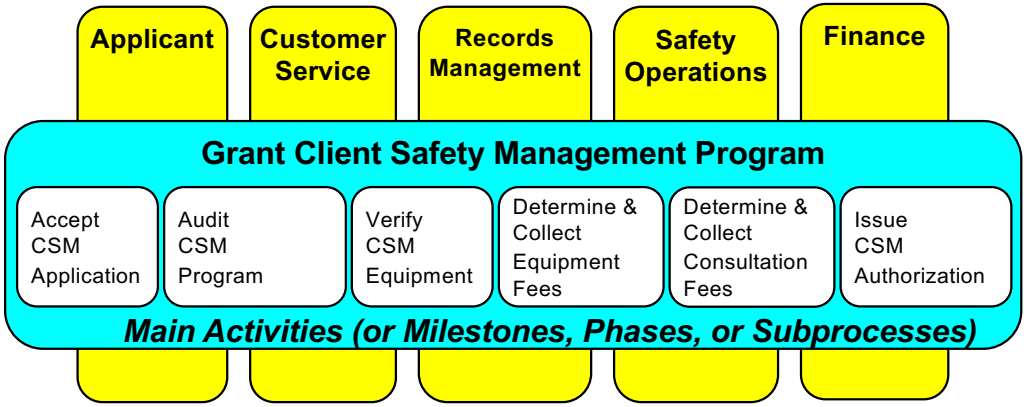
Client Result:
Approval granted for a self-managed safety program.

Agency Result:
Revenue collected.
New participant in CSMP; confirmation that regulations are satisfied

Cases:

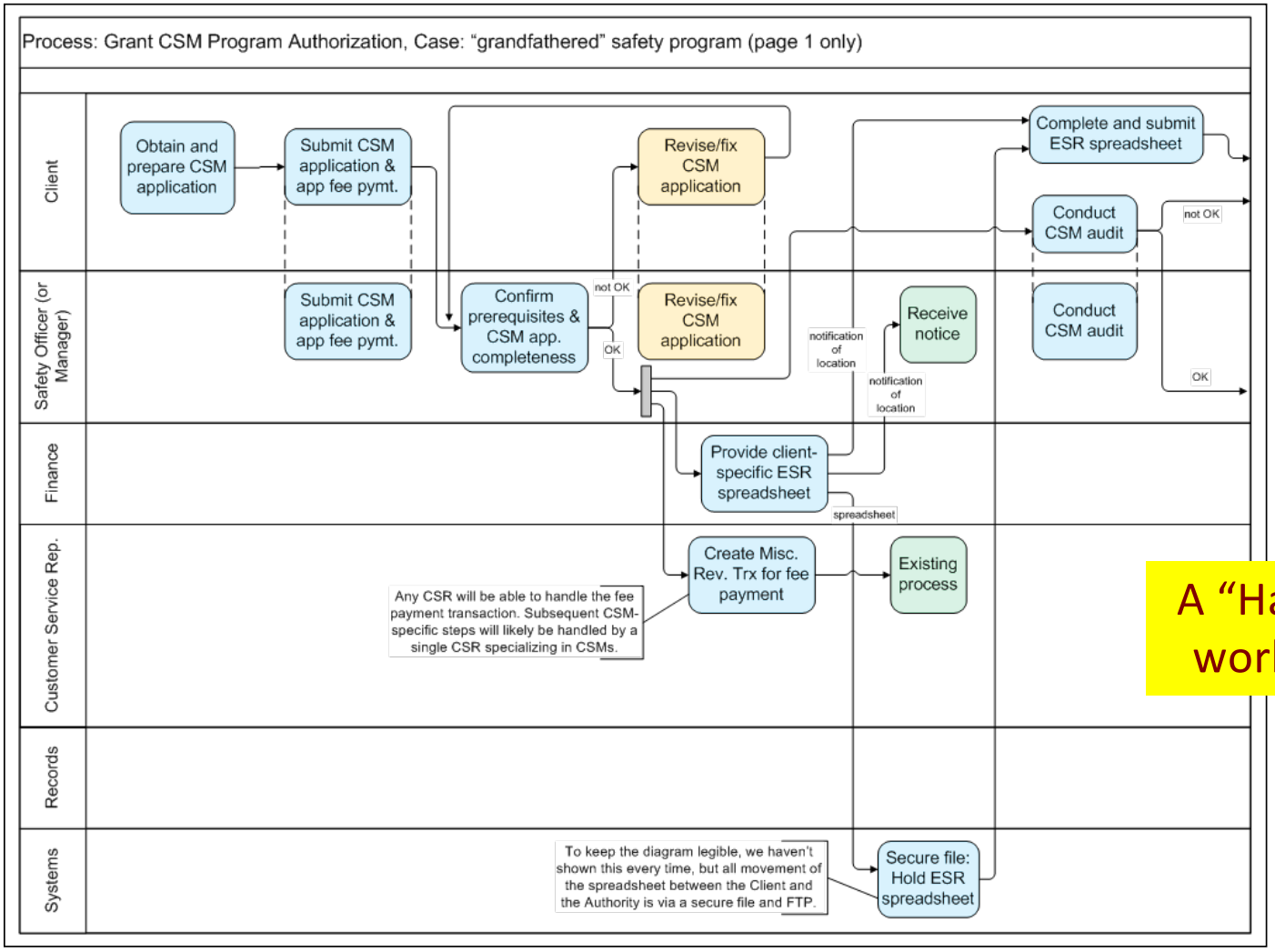
- New
- Grandfathered
- Ownership Change

Process Scope Model – pure "what"…



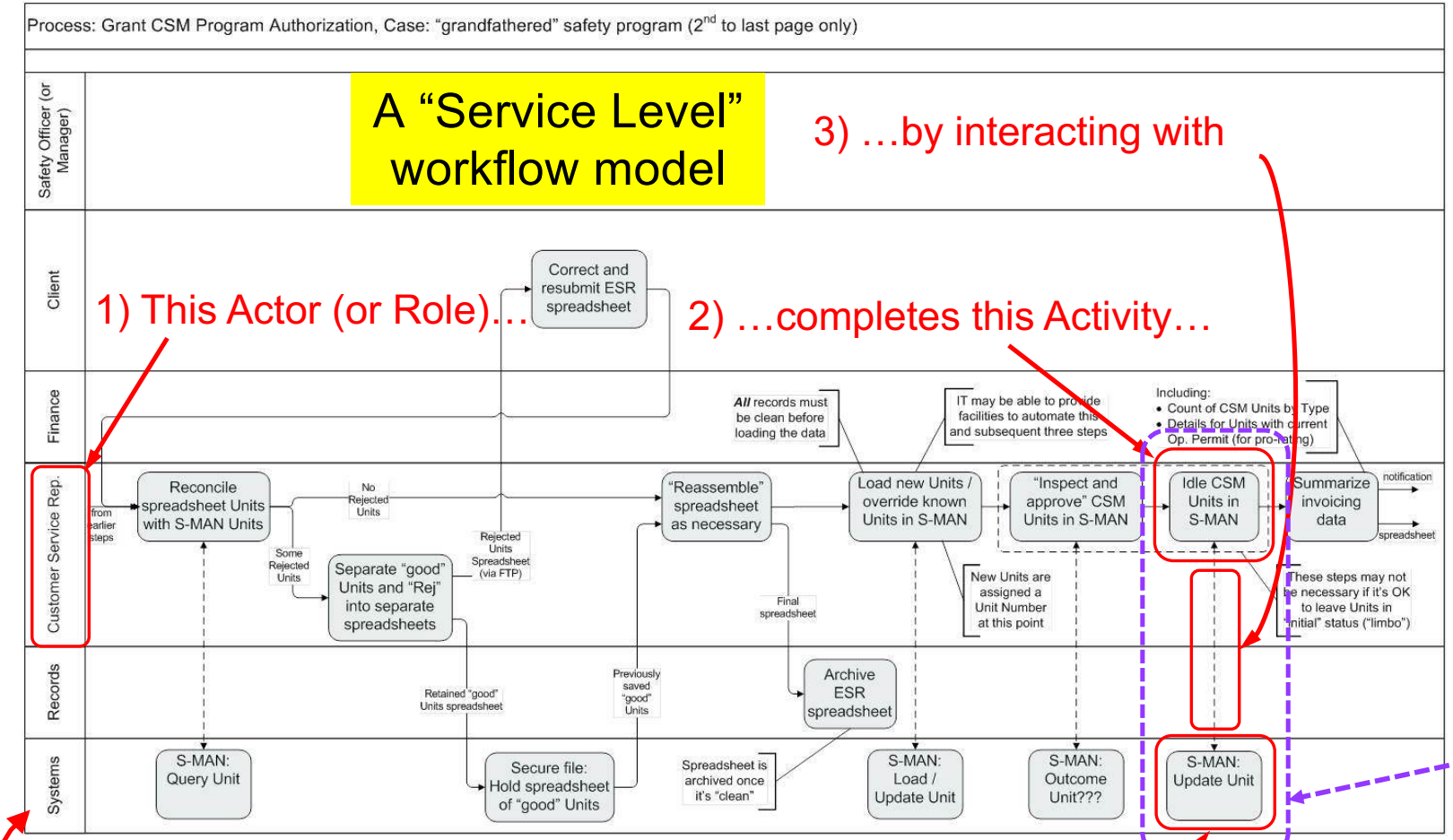
Process Summary Chart – simplified "what," plus "who"

The initial, business-friendly workflow model



A "Handoff Level" workflow model

Then detail showing where use cases & services fit

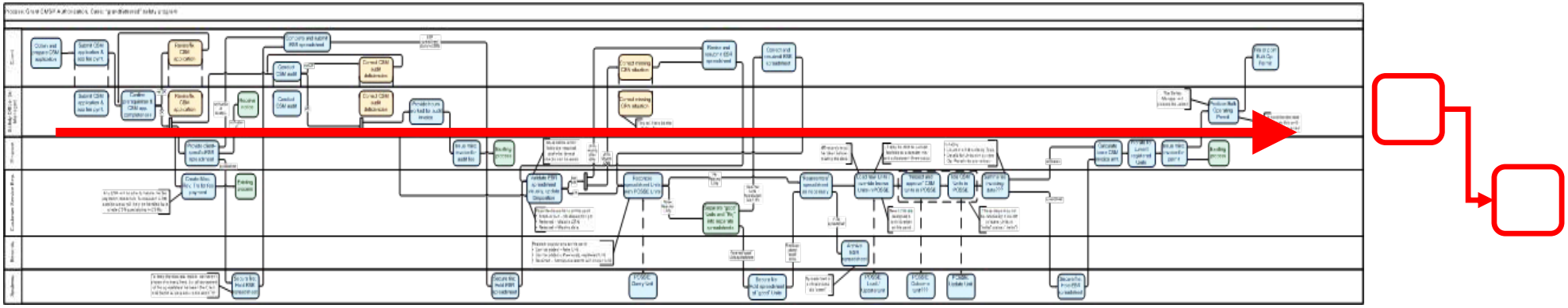


That's a Use Case!
- an actor
- interacting with a system
- to obtain a service
- to help them complete a task or obtain information

is what we mean by a Use Case (which may begin as a User Story)

Mission accomplished! Conclusions:

- "Plan A" rejected – agreement that Unit data *must* get into S-MAN
- “Plan B” (change the app) looks good, but the vendor estimates are *HIGH*
- “Plan B Minus” (existing functionality plus CSR work) is *worth the cost*



1. If requirements, issues, assumptions, etc. are in lists, people will argue endlessly; if they are in an *integrated* and *understandable* set of models, it’s much harder to dismiss the reality of the situation
2. Process Models, Use Cases, Service Specs, & *Concept Models*: ***essential!***

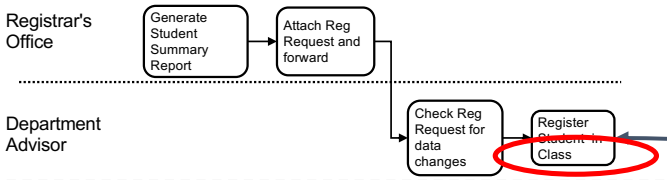
Everything relies on the Concept Model / Data Model

All use the language and constraints of the Concept Model (the "thing model") – the ultimate "what"

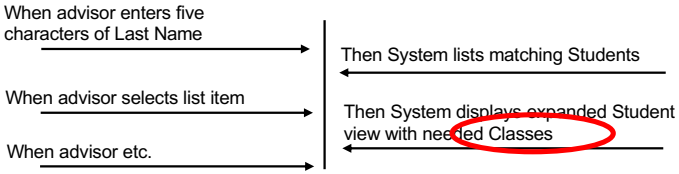
Business Goals & Objectives

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.

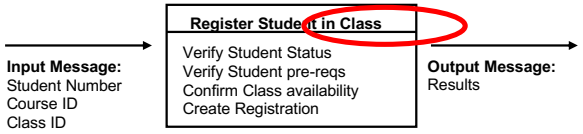
2 Business Process



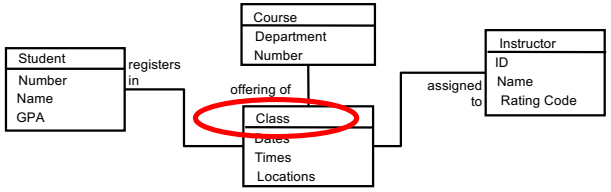
4 Presentation Layer (user interface)



3 Application Layer (rules & logic)



1 Data Layer (data & storage)



My usual sequence

Use Case
actor + service + platform:
Advisor *Register Student in Class* via SRS

Service
verb + noun (+ noun):
Register Student in Class

Entity
noun:
Class

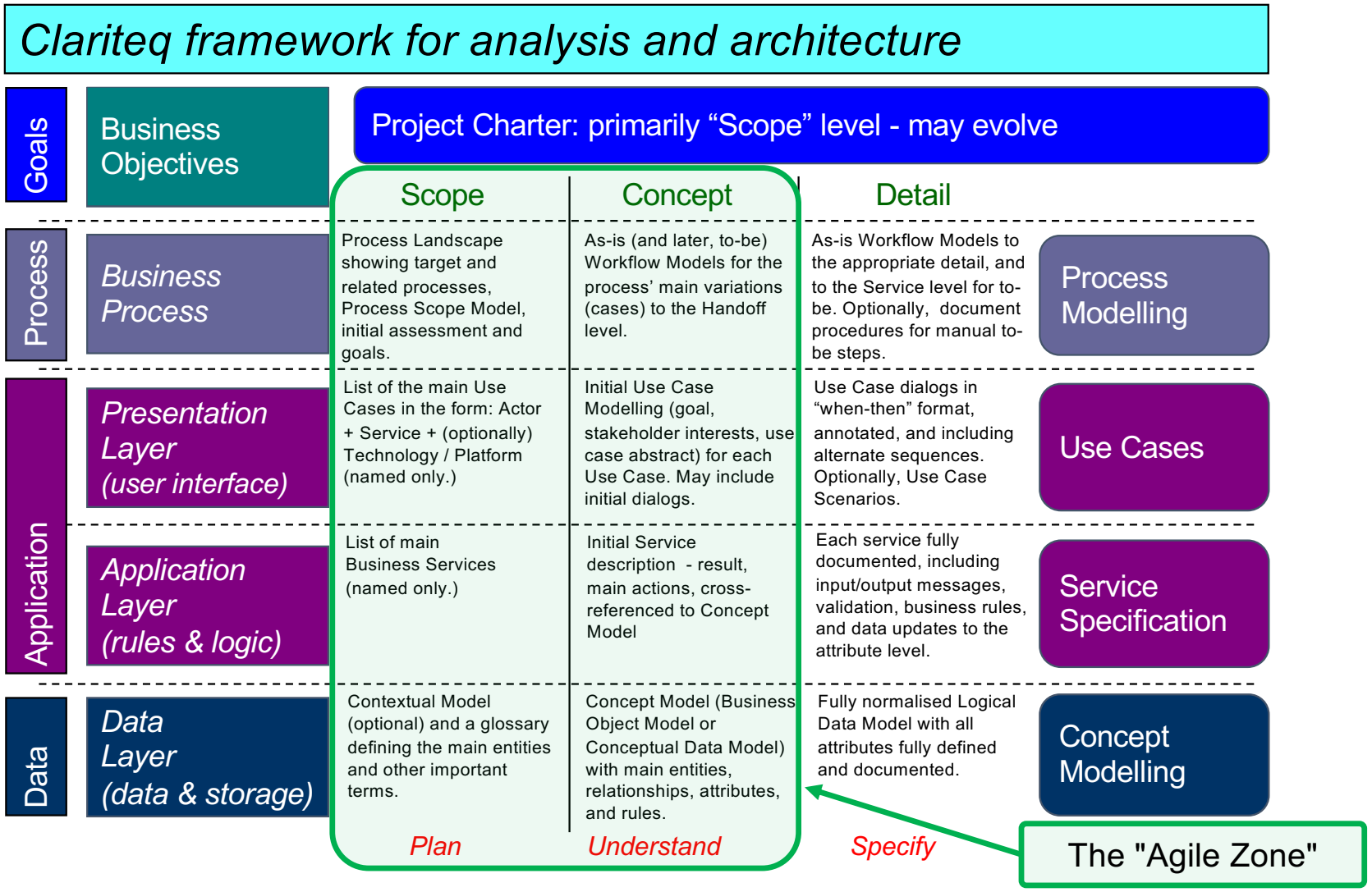
Use Cases/User Stories:
- Who (Actors) needs access to the Services, and how (Platform)?

Verb-Noun pairs:
- The *Services* (event-handlers) that are at the heart of a *Service Oriented Architecture.*
- *Also* "building blocks" of Business Processes

The core *Nouns or* Things in your enterprise. Also known as *Business Objects.*

Bonus – great starting point to discover your Events/Services and Use Cases/User Stories

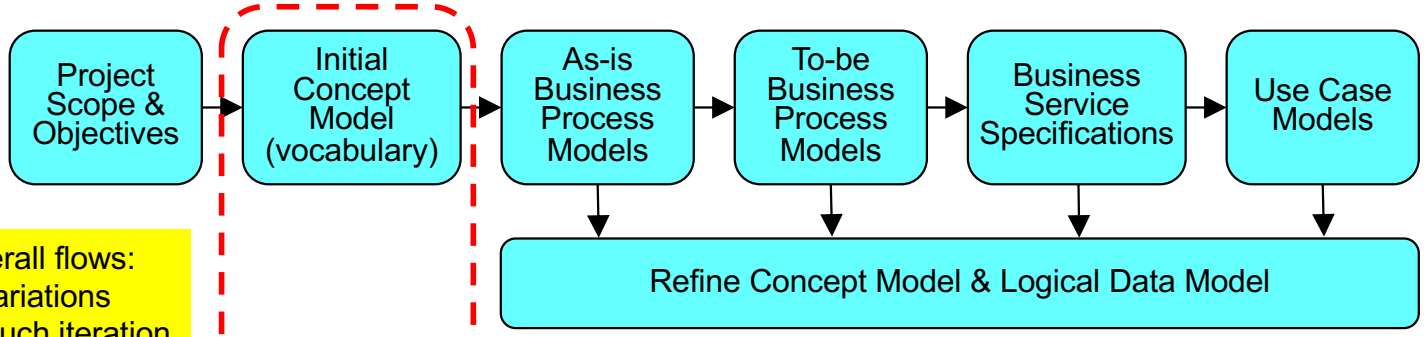
With progressive detail, Concept Modelling supports Agile



Techniques and methodologies

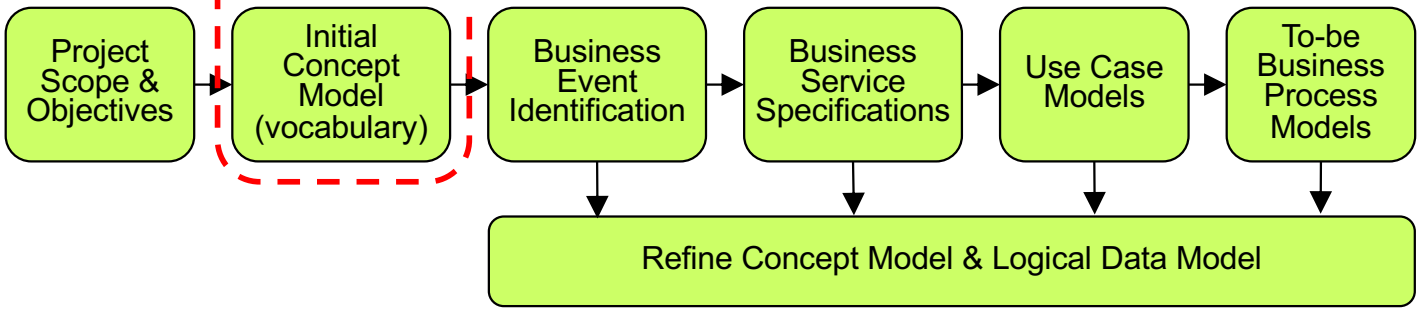
- The same techniques are used in different sequences, with different emphasis, in different situations.
- Concept Modelling to clarify language is a great starting point.

Larger project (enterprise-wide, cross-functional): process-oriented / "outside-in" –



These are typical overall flows:
- there are many variations
- there is always much iteration

Smaller project (local, departmental): service or use case-oriented / "inside-out" –



This slide left blank by accident…

Business Process Fundamentals

Course Topics

1. Requirements Definition
   - Goals, Issues, and the Return of Modelling
   - Case Study - Integrating the Techniques
2. Business Process Fundamentals
   - Five Things You Need to Know
   - Discovering, Scoping, & Assessing Your Processes
3. Concept Modelling Fundamentals
   - E, R, A - A Concept Model's Essential Components
   - Drawing Your Model for Maximum Understanding
4. Business Process Workflow Modelling & Design
   - Five Core Guidelines for Great Swimlane Diagrams
   - Facilitating a Process Mapping Session
   - Assessment of the As-Is and Transition to the To-Be
5. The Process-Data Connection
   - The Natural Synergy between Process & Data Models
   - Process-Data Synergies in Modelling, Analysis, & SW

Five central ideas

1. It's essential to have clarity on what a *business process* really is

2. Existing performance measures are often *functionally aligned* and work *against* business processes

3. Enterprise system implementations *must* include a *business process* perspective

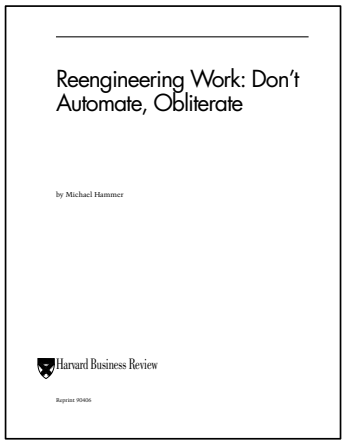
4. Success with business processes depends on taking a *holistic view* in which six *enablers* are considered

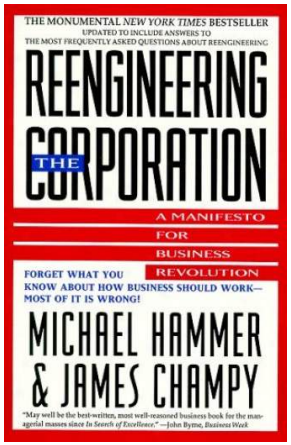
5. Business processes can't be great at everything – a single *differentiator* or *strategic discipline* should be chosen

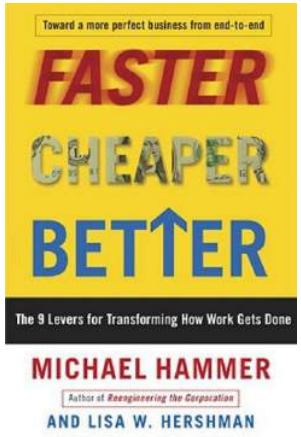
1. Confusion – what **is** a "business process?"

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. Enterprise system implementations must include a business process perspective
4. Success with business processes requires a *holistic view* in which *six enablers* are considered
5. 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*)

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

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

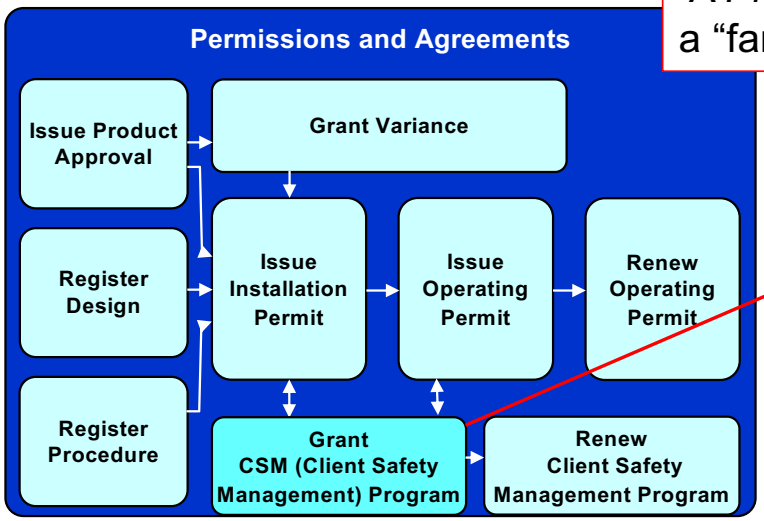
Seek balance – a "business process" lies between the extremes

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

The key issues – granularity and orientation

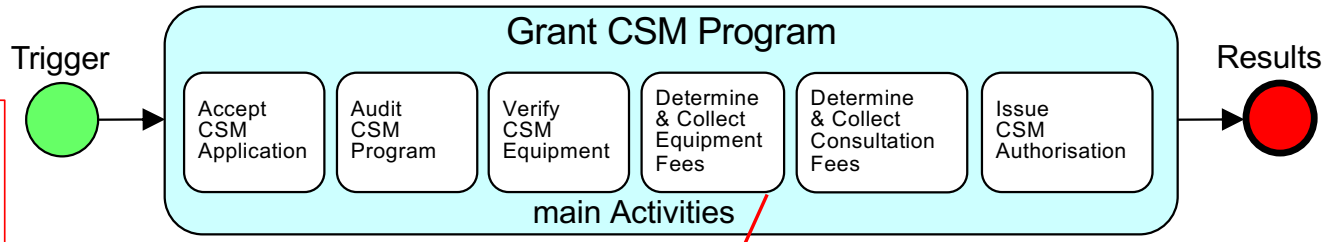
Taxonomy: a collection of processes vs. a process vs. a procedure

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*

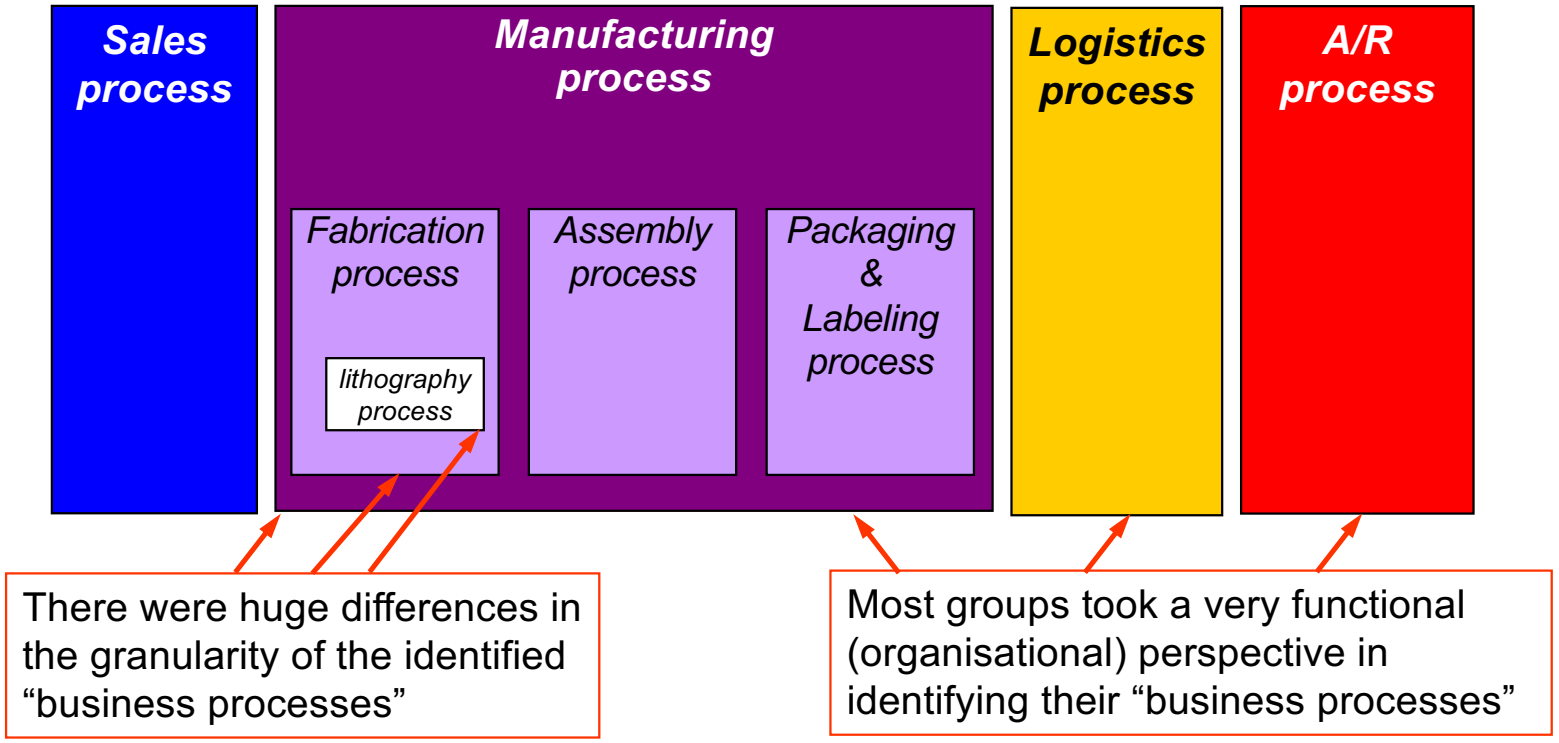
Procedure:
A set of step-by-step work instructions
(a job aid) for a specific task or activity
that will yield identical results every time

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…

A real life (and expensive!) example

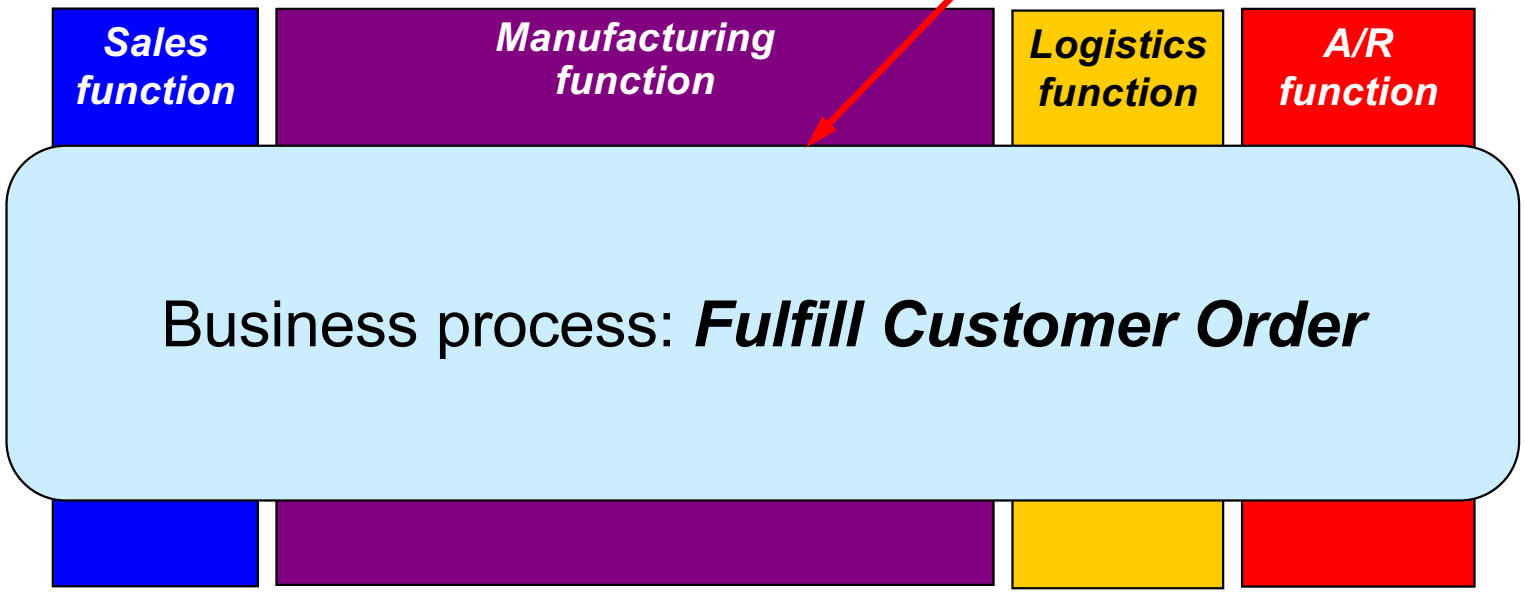
As part of a massive system implementation, a global manufacturer identified the *business processes* that were expected to improve:



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.



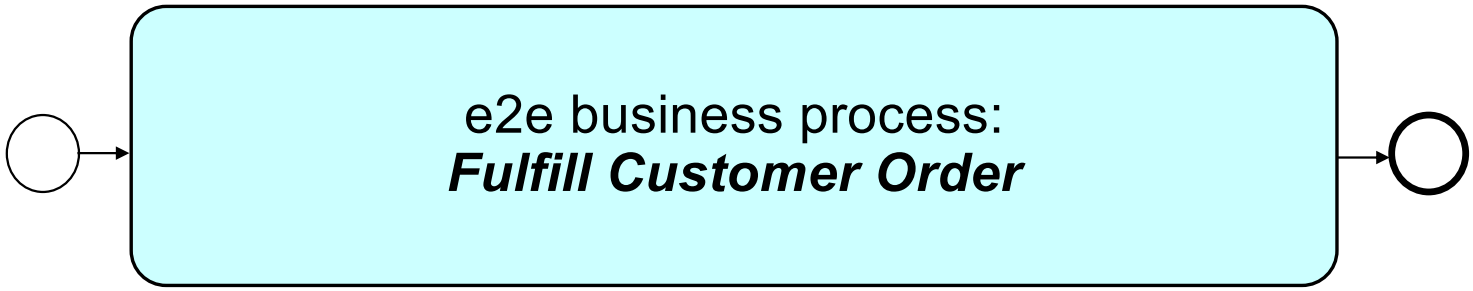
"Business Process" =
end-to-end, cross-functional, business process.
"Larger" than people think – from *initial* trigger to *final* results.

Discuss - what are the boundaries of the process?

Trigger

Result

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.*

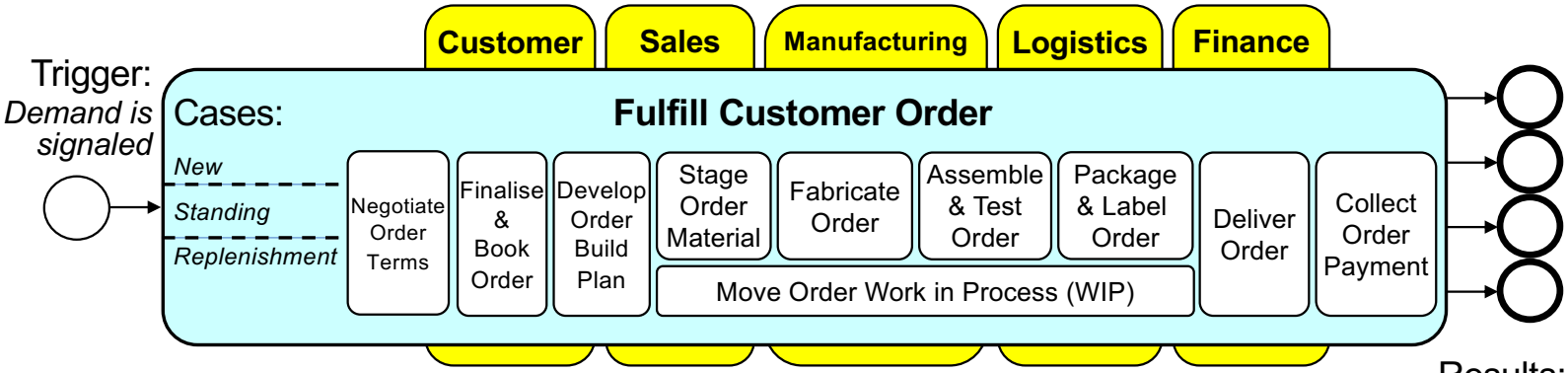
Order is Received, Tested, and Accepted? *Yes.*

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

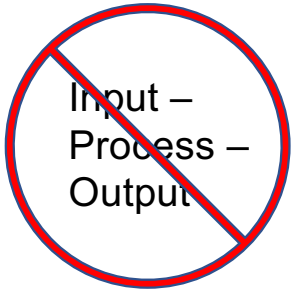
Industry Association:
Order stats reported

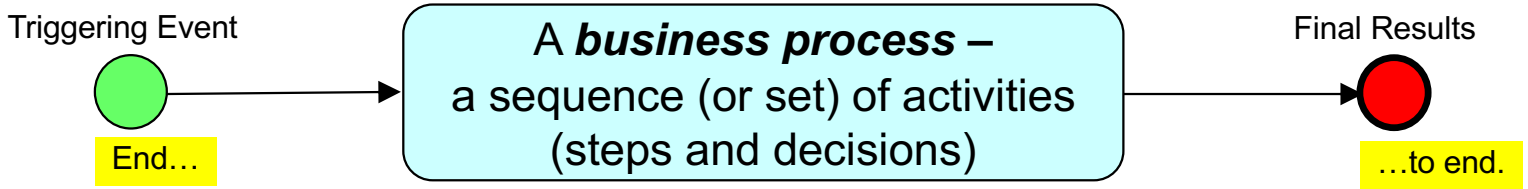
Always construct a *Process Scope Model* & a *Process Summary Chart* before diving into Workflow Modelling / Swimlane Diagramming

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"

Naming conventions will make life easier

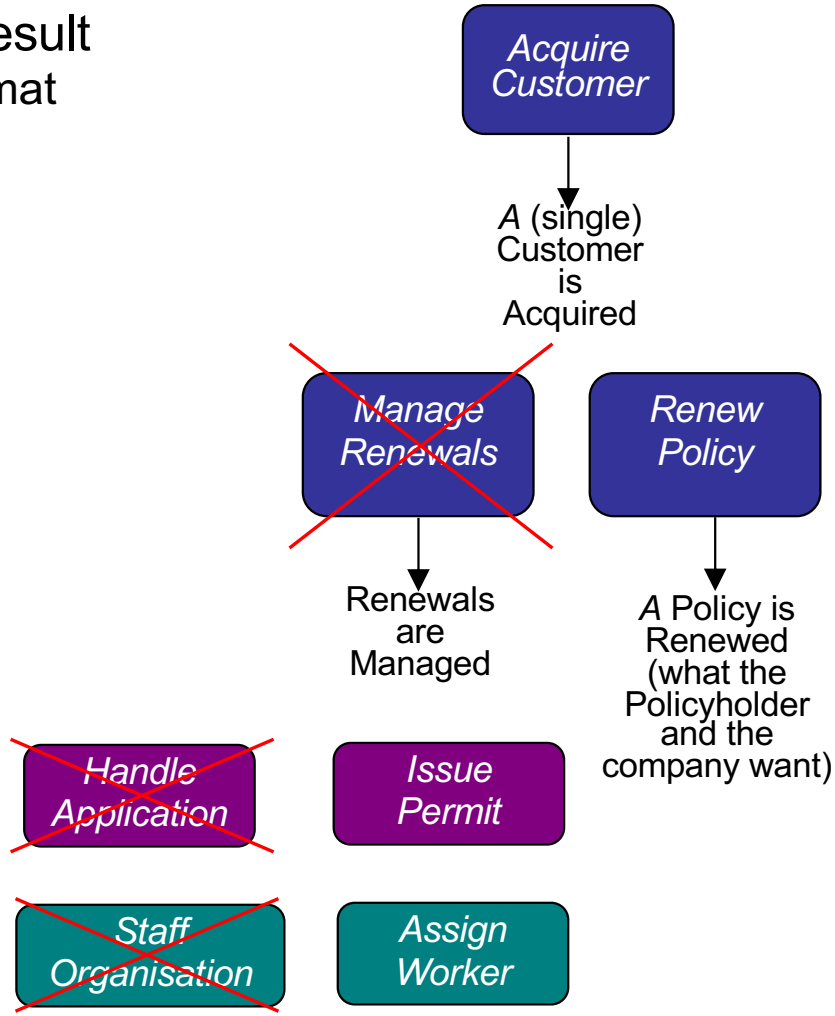
1. The process name ***must*** indicate the expected result
   - Name potential process in “active verb – noun” format
   - Restate that name as a result (“noun is verbed”)
   - Ensure this is the intended result of the process: *discrete*, so results are *identifiable* & *countable*

   - ***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?)

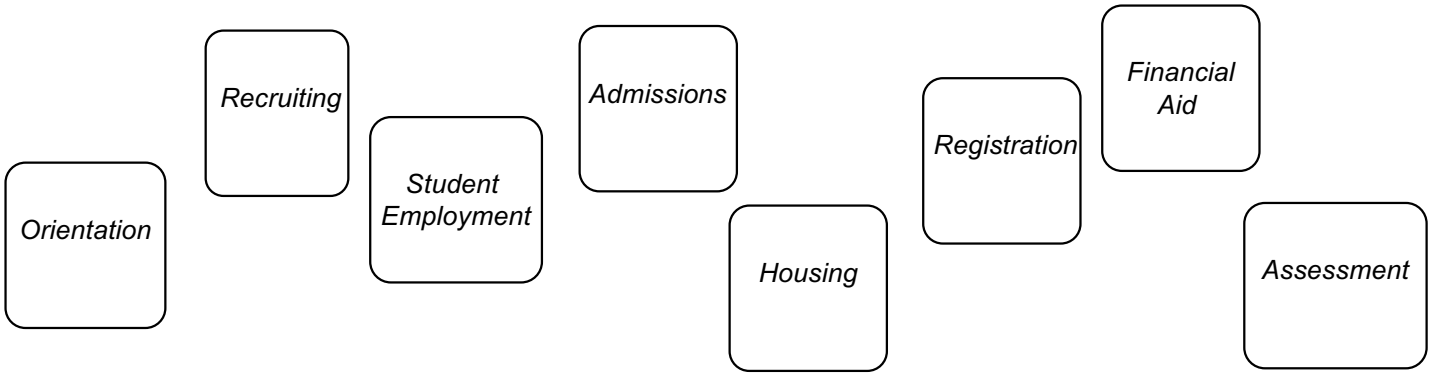
3. Name process in the singular



An example from higher education

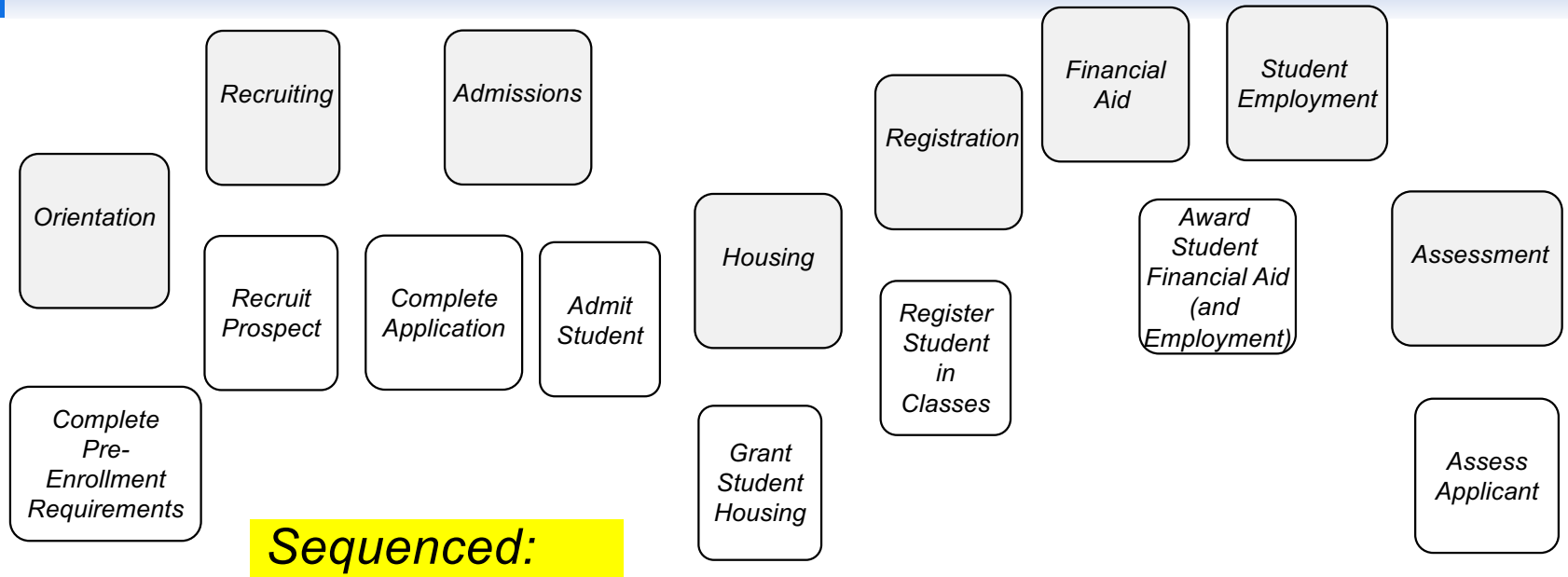
As part of a strategic initiative to address falling graduation rates, a university took a process-based approach to determine why they were failing to admit the most promising candidates…

The "processes" that were initially identified…

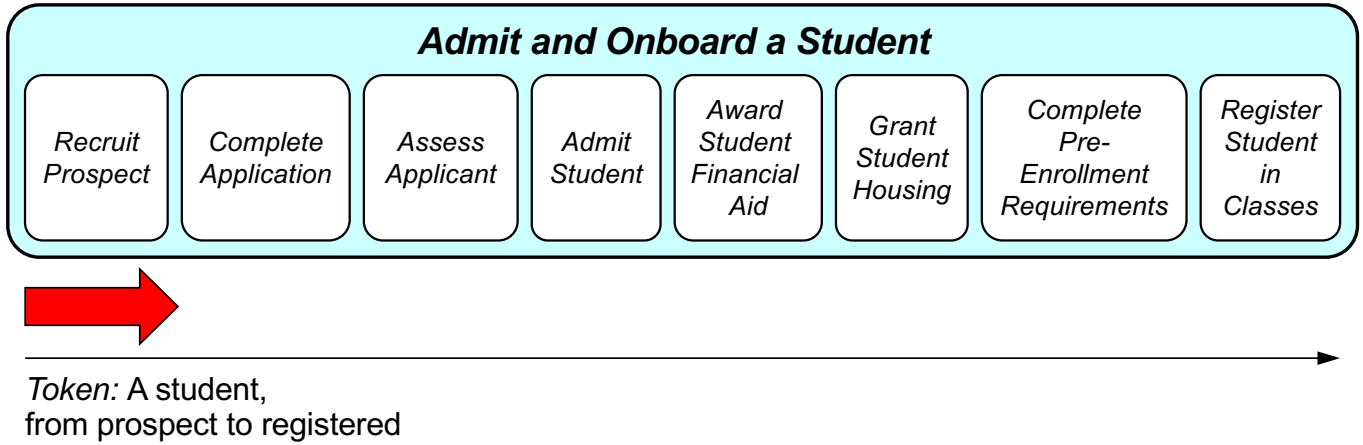


Are these good business processes?
NO! Each of these is a department or function.
We convened a facilitated session to determine the "real" process

Rename, reduce, refine, and sequence



Sequenced:



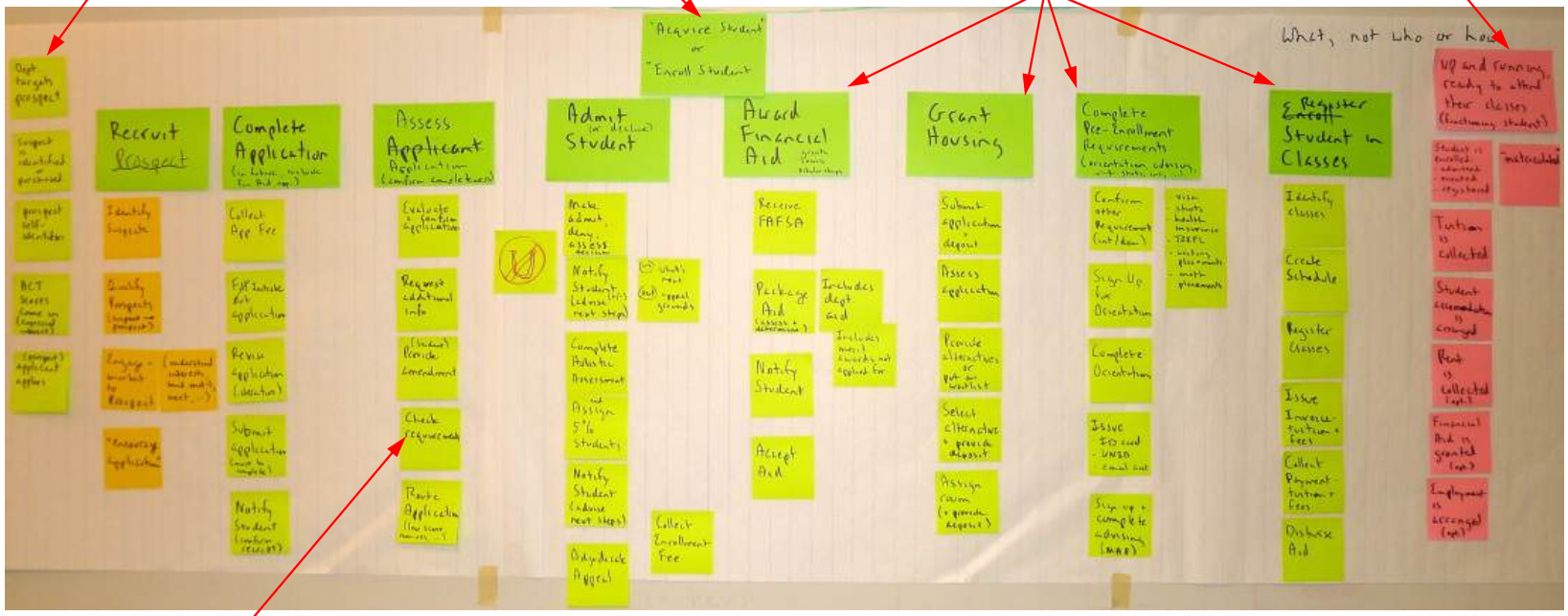
From the session – "Is it a single X-functional process?"

Triggers

The process

major Activities (or phases, milestones, subprocesses…)

Results



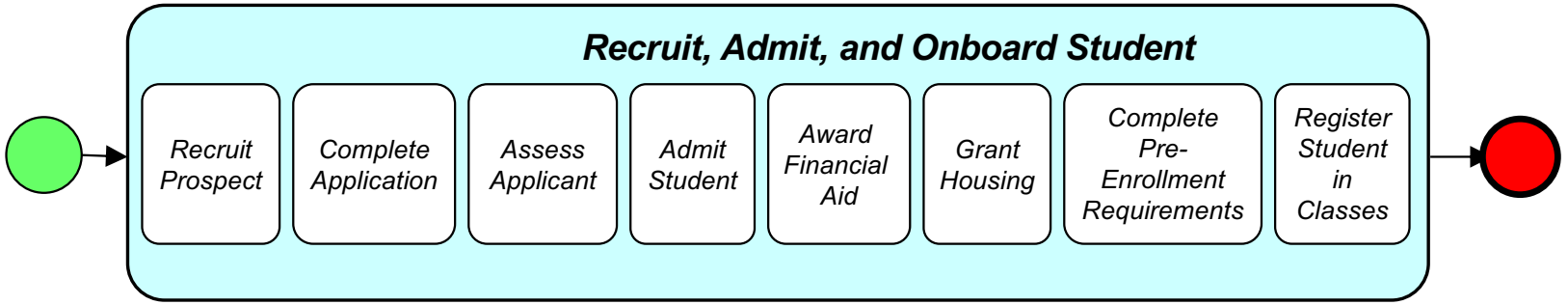
More detailed activities

Focus is on "what, not who or how."

Note the high-tech tools.

Very iterative, but only 90 minutes!

The cleaned-up "Process Scope Model"



Triggering Events:
- Dept. targets prospect
- Suspect is identified or purchased
- Prospect self-identifies
- ACT scores come in
- Prospect applies
- …

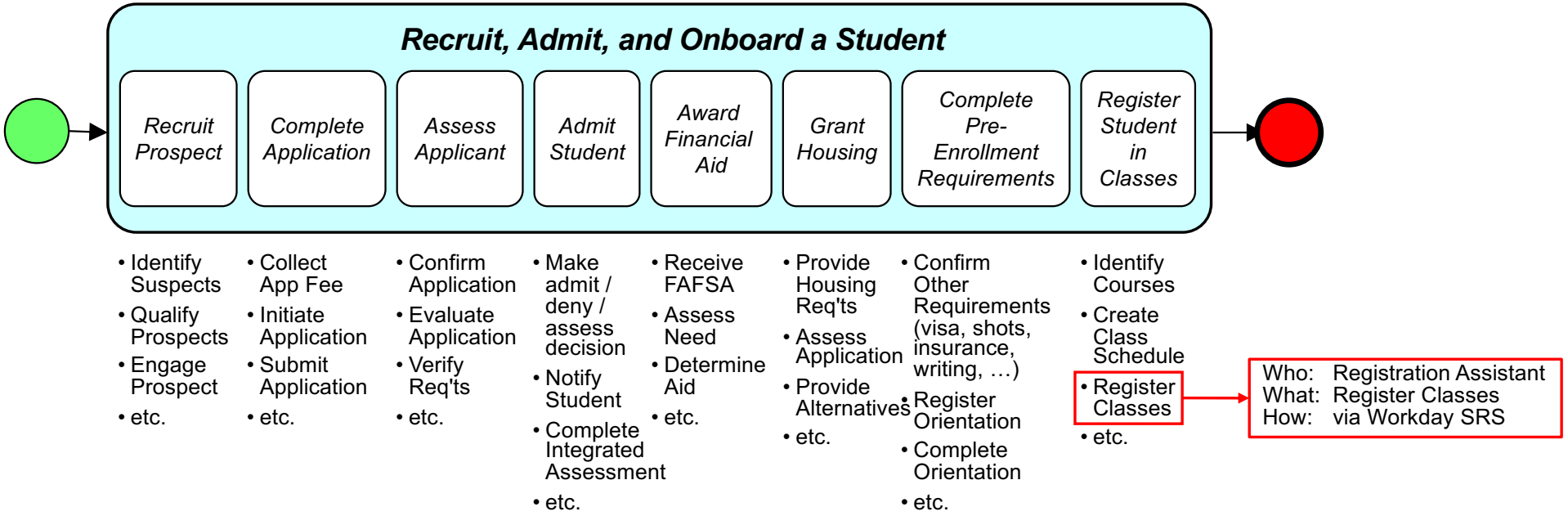
Cases:
- In-state undergrad
- Out-of-state undergrad
- …

Final Results:

"Up and running," ready to attend classes:
- Student is:
  - admitted
  - oriented
  - registered
- Tuition is collected
- Student accommodation is arranged
- Financial aid is granted
- Employment is arranged
- …

TRAC –
- Trigger
- Results
- Activities (~5-7 phases or milestones)
- Cases (major Variants)

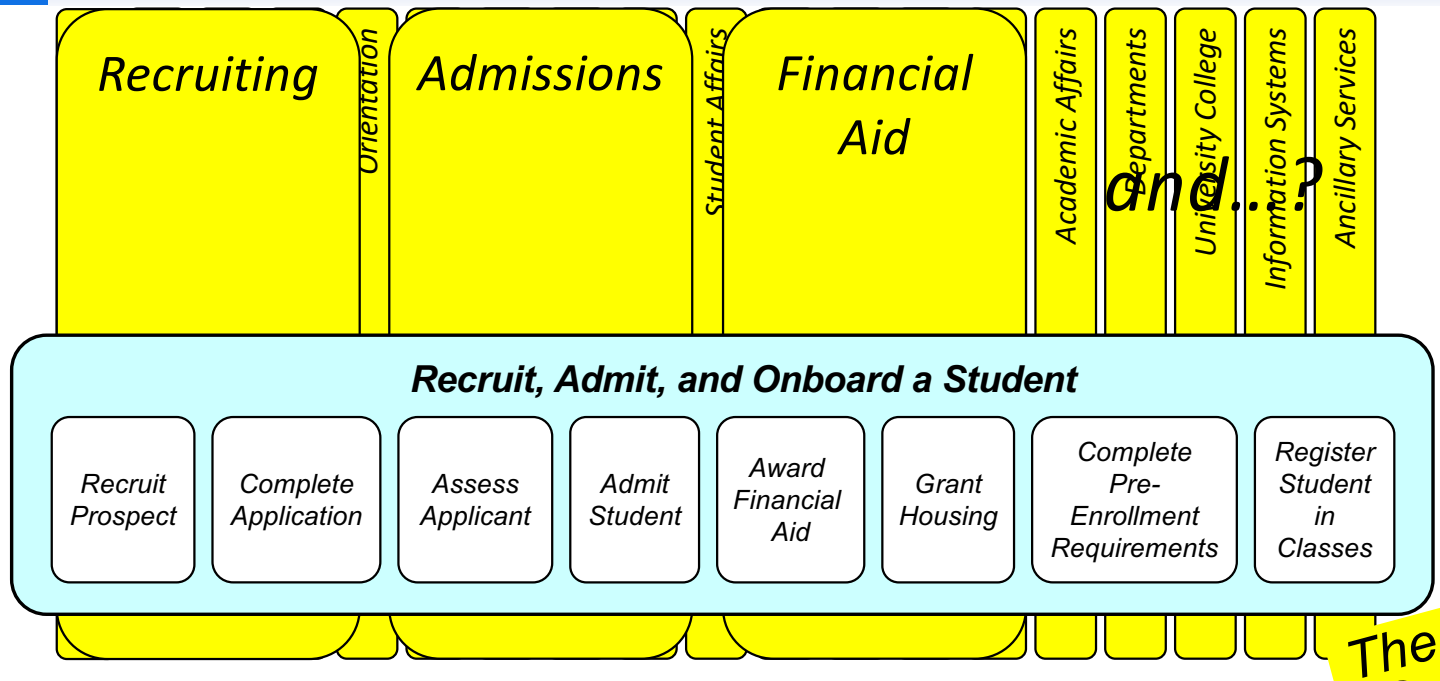
The cleaned-up "Augmented Scope Model"



Typically, 5 – 7 activities identified within each major activity.
Initially just "what" (verb – noun) – later, add "who and how,"
e.g., Registration Assistant (who) Register Classes (what) via Workday SRS (how)

Identifying the functional area responsible for each activity revealed the process was massively cross-functional…

Process Summary Chart shows an astonishingly cross-functional process



The point – the execs said "Get on with it! There's no need to burn up $50,000 on a business case."

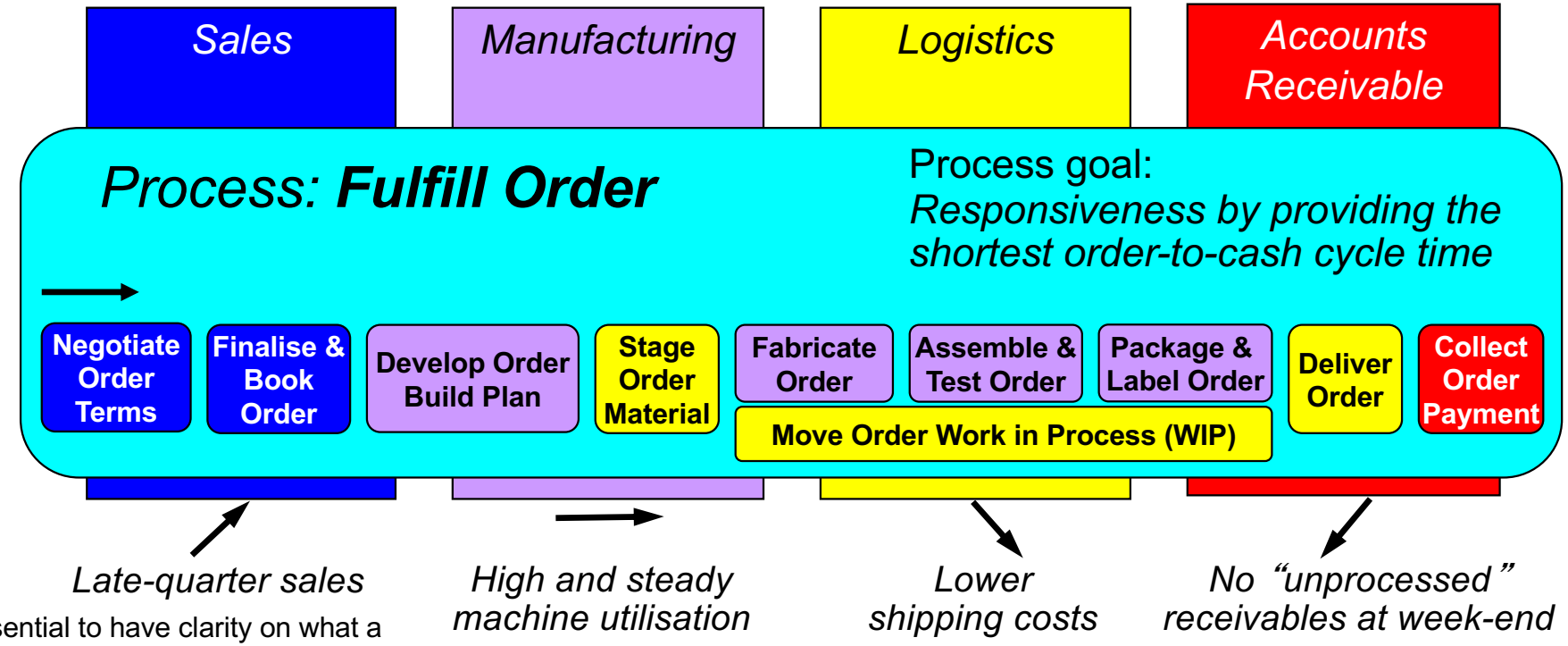
Without explicitly addressing the end-to-end process:

- almost no chance the student experience is positive
- very frustrating for the people doing the work
- almost no chance the university is going to meet its goals

Two key points:

1. *Functions are doing their best to optimise their activities*
2. *A multitude of dis-integrated systems and data sources are being used*

2. A common obstacle – misaligned performance measures

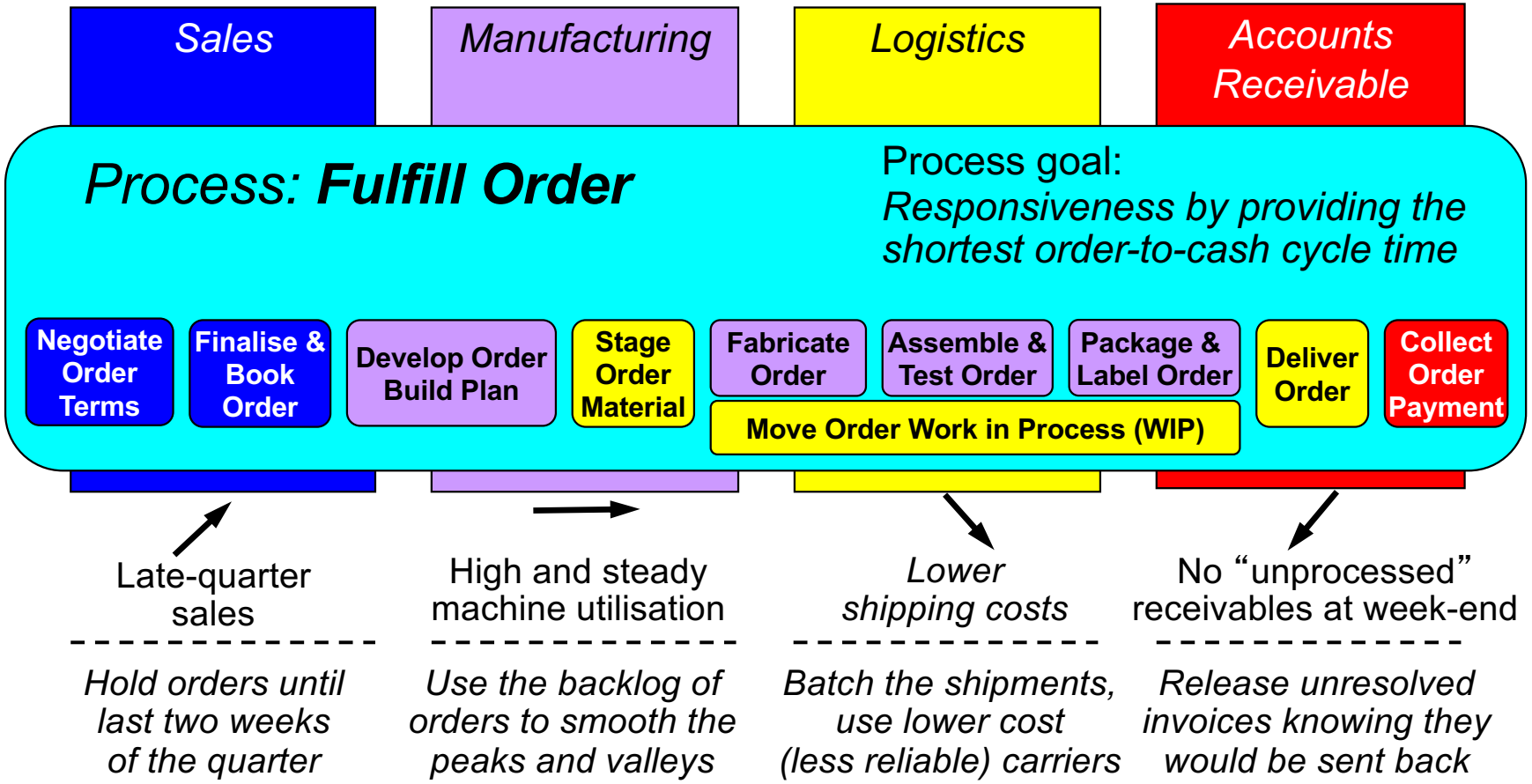


1. It is essential to have clarity on what a *business process* really is
2. Performance measures may be *functionally aligned* and work *against* business processes
3. Enterprise system implementations must include a business process perspective
4. Success with business processes requires a *holistic view* in which six *enablers* are considered
5. 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*

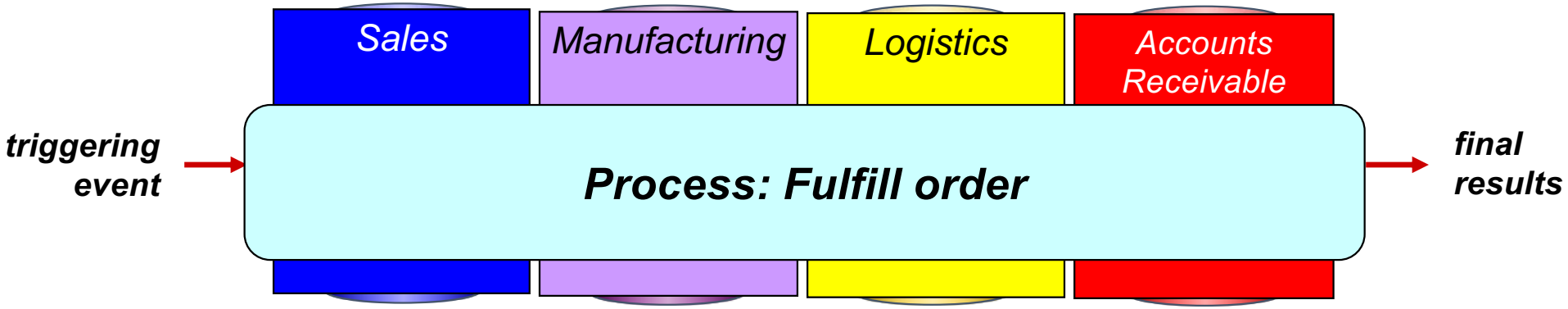
Discuss –
What are the likely impacts of these performance goals?
What will the different functions do to meet the targets?

Misaligned performance measures



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

This doesn't mean functions are bad!



Process

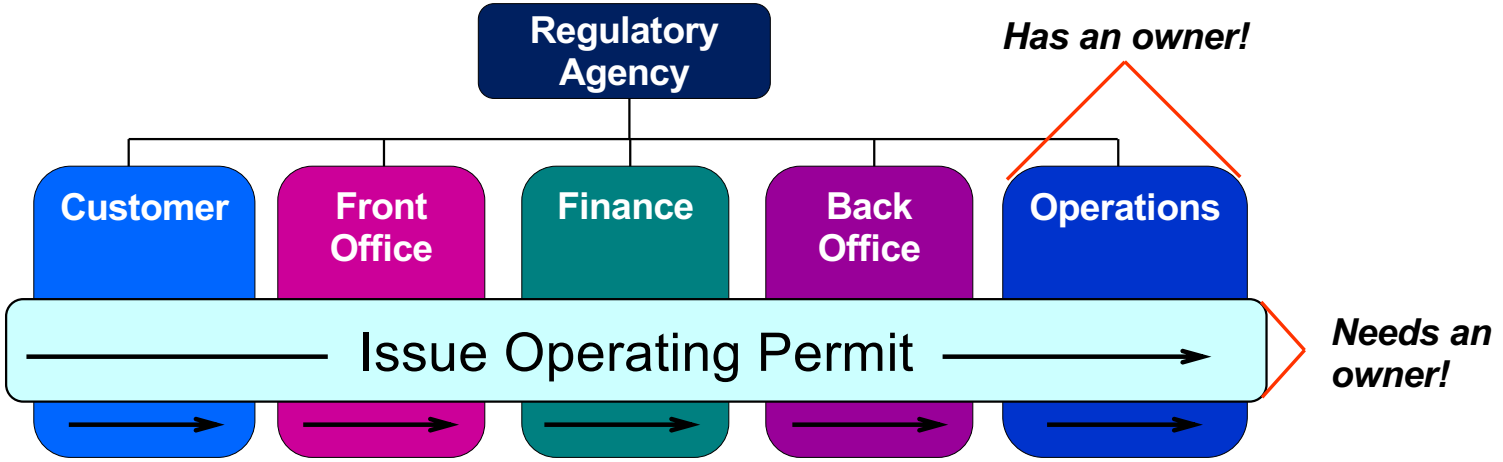
- End-to-end business processes deliver ***essential results*** by aligning the work of multiple functions
- Must be identified and managed as a whole

Function

- A ***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

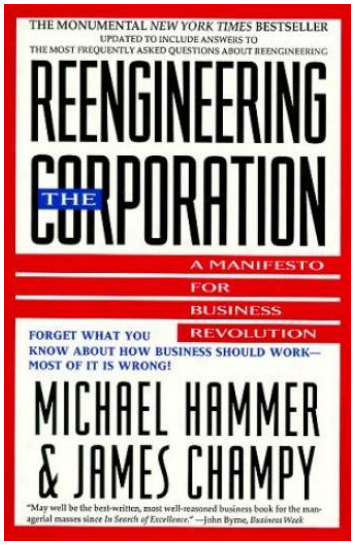
It takes concerted effort – nothing happens by accident

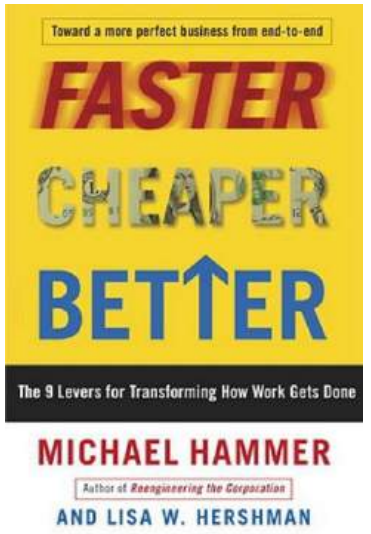
3 – Processes and information systems

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. Enterprise system implementations must include a business process perspective
4. Success with business processes requires a *holistic view* in which *six enablers* are considered
5. A business process can't be great at everything – a single *differentiator* must be chosen

"Success with SAP Implementation"

Study by the late Michael Hammer, "godfather of BPR"



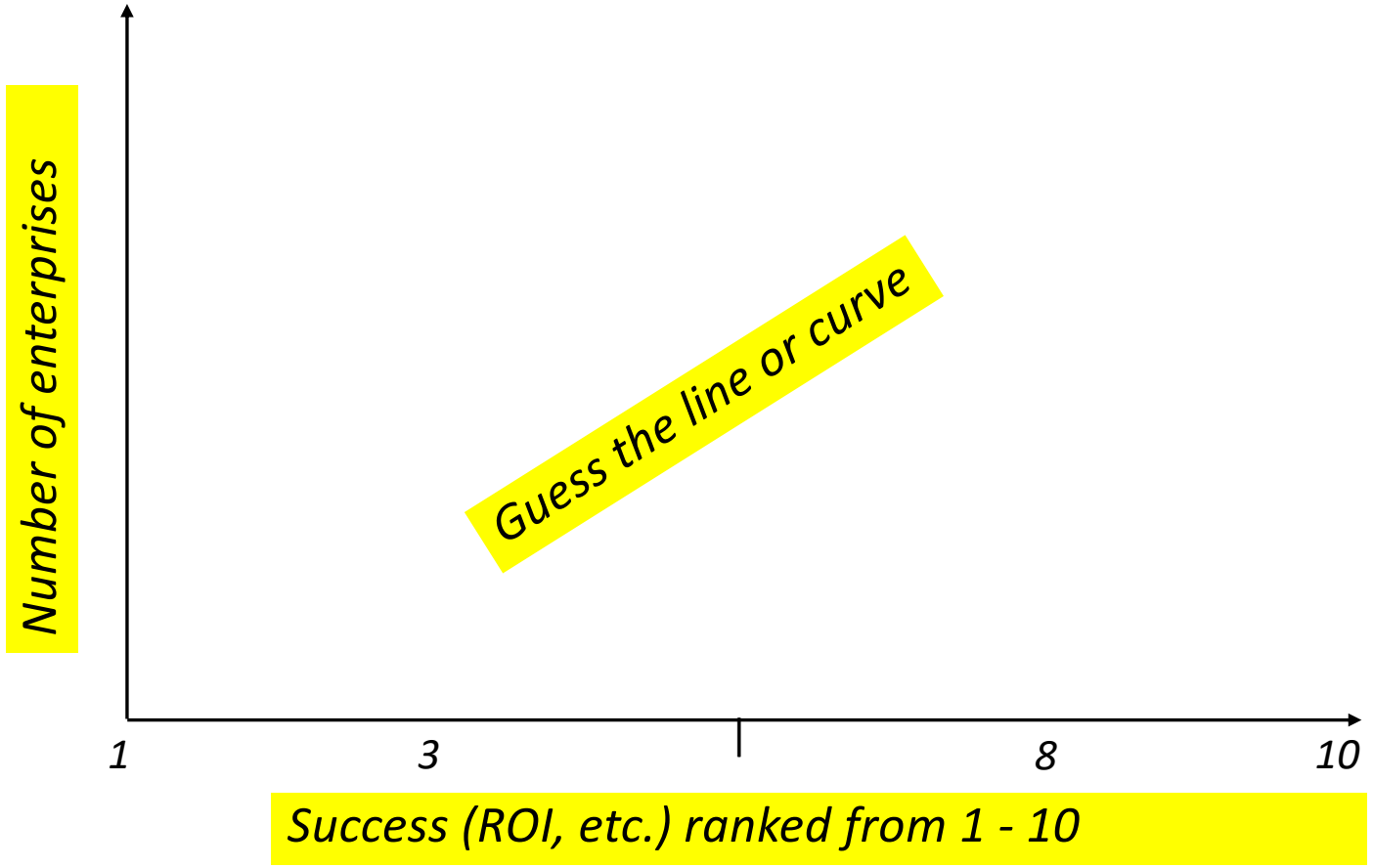


Observed that success of SAP implementations varied *wildly*

Worked with ~80 companies to assess their degree of success with SAP implementation

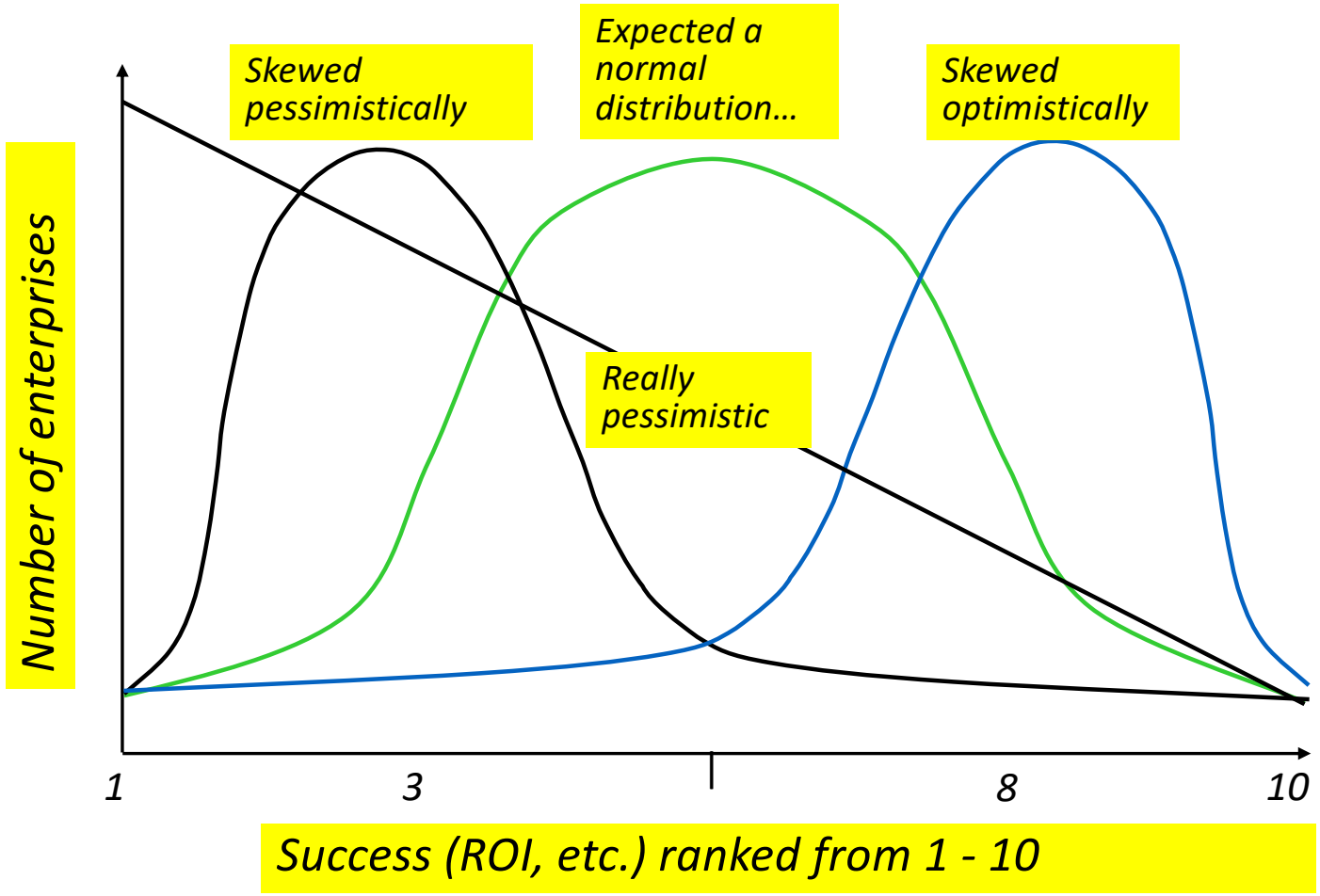
Success with SAP implementation

Hammer plotted the number of companies for each “success” ranking

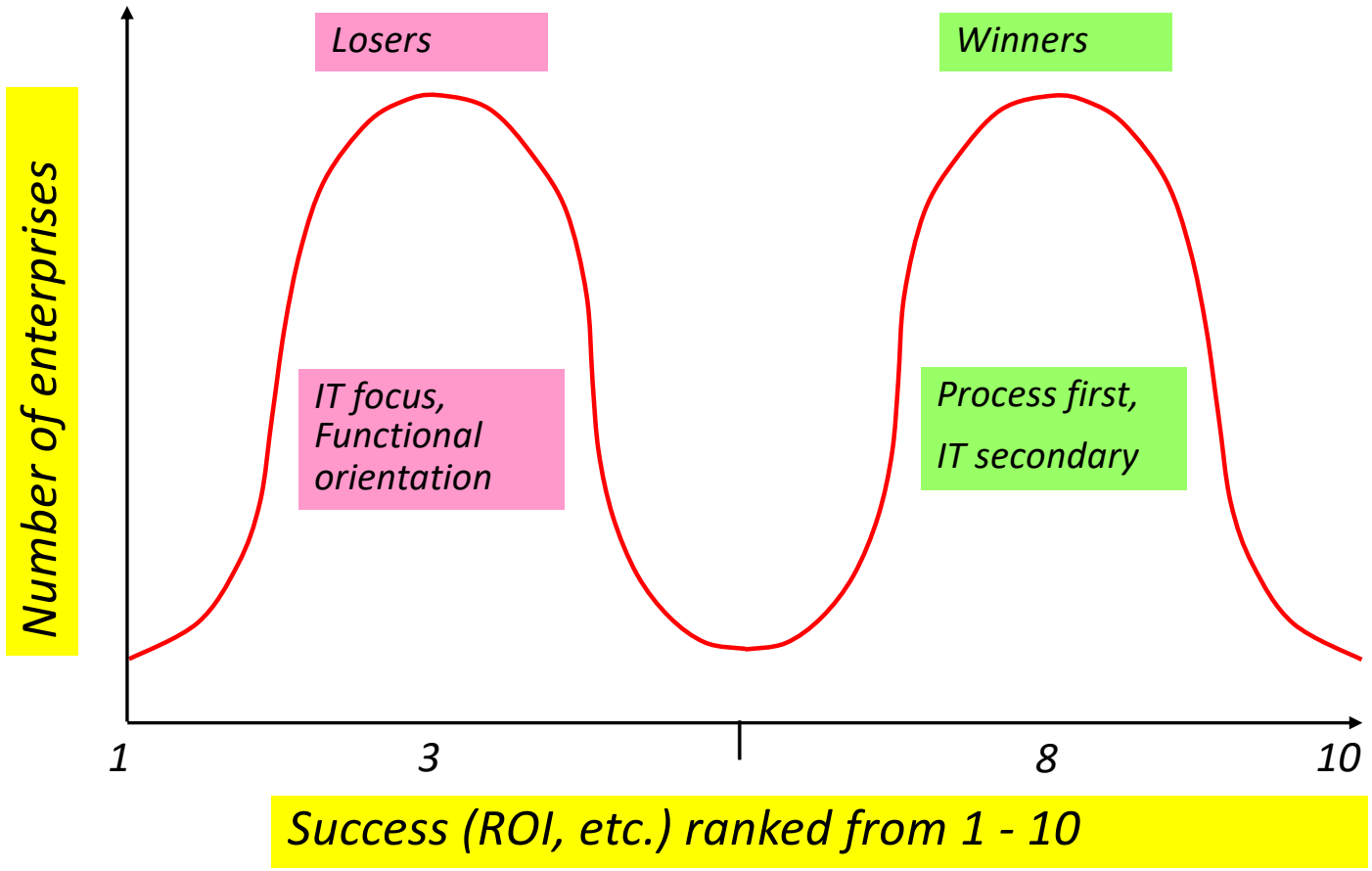


60

Hammer not sure what the outcome would be



The surprising result



Returning to an earlier example

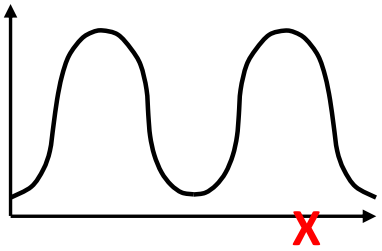
Global manufacturer implementing SAP

Four primary modules:

- *Sales*
- *Manufacturing*
- *Logistics*
- *Finance*

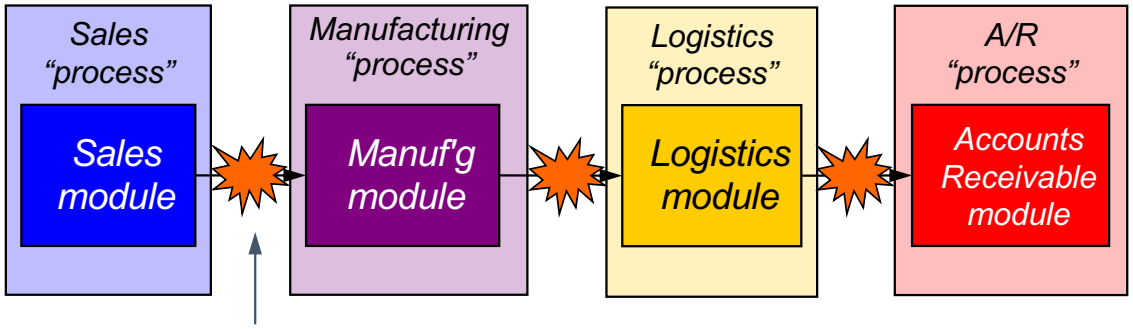
Determined to *do it right:*

"This will be a *process-oriented* implementation!"



Outcome and resolution

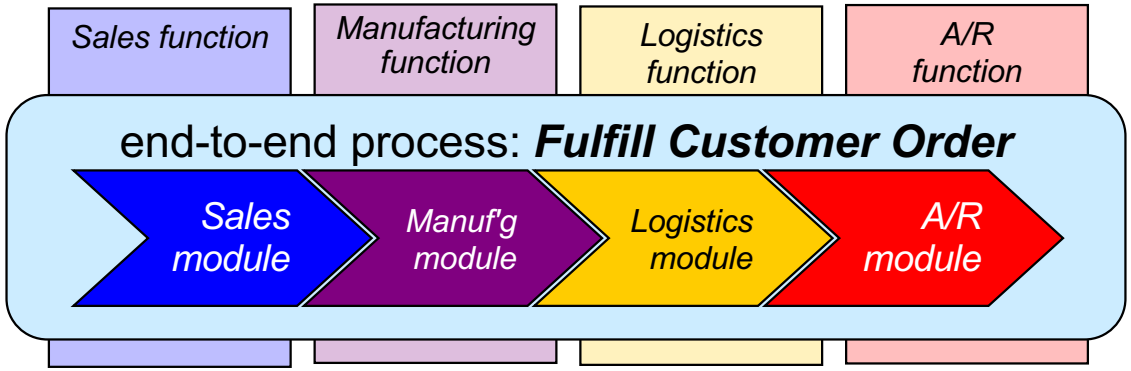
SAP was implemented without clarity on "process" –



Poor performance

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

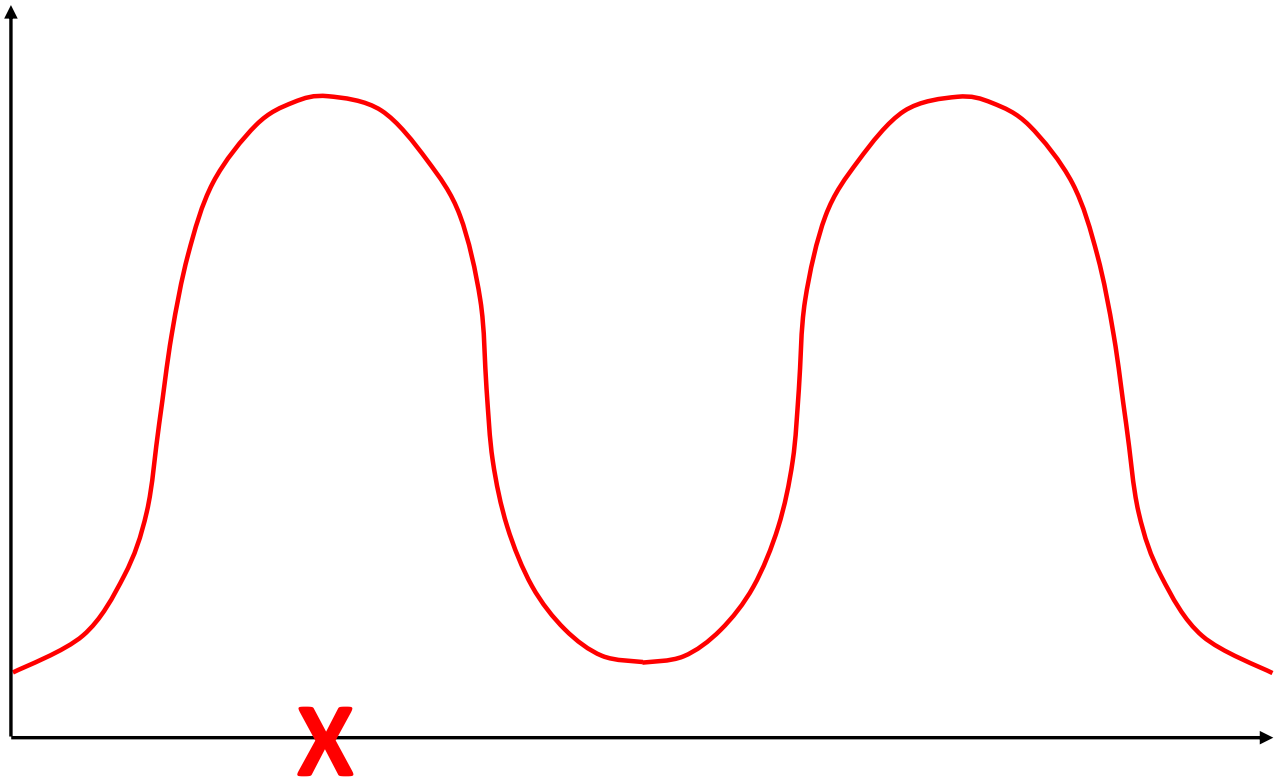
SAP re-implemented in a process-driven configuration –



Great performance

Same software, radically different outcomes

Staying “right” in an “entropic” environment



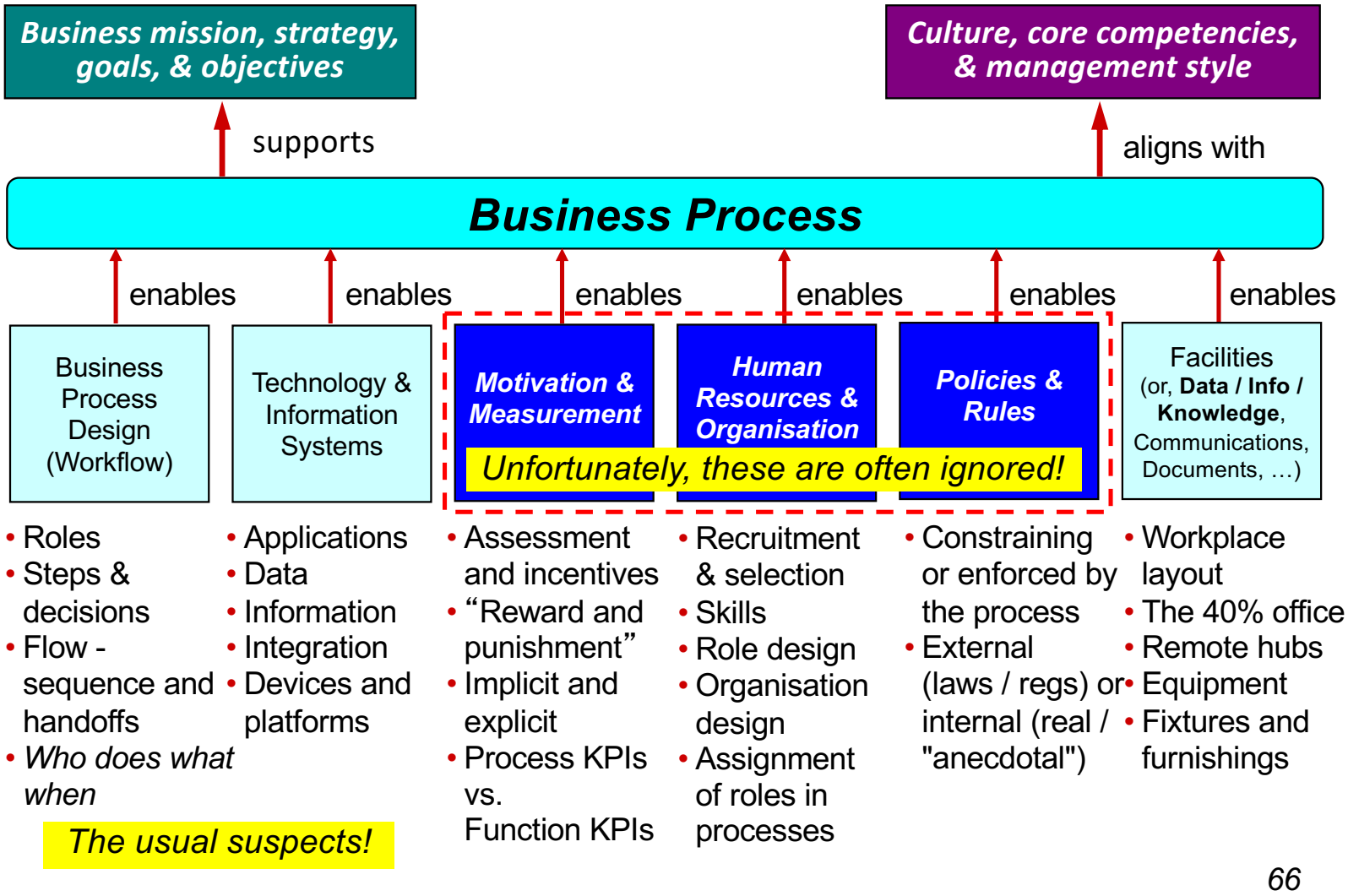
There will always be a pull back towards functional comfort

- *ongoing management of the process is critical!*
- *all enablers must be addressed for a sustainable process*

4. A holistic view for process analysis and design

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. Enterprise system implementations must include a business process perspective
4. Success with business processes requires a *holistic view* in which six *enablers* are considered
5. A business process can't be great at everything – a single *differentiator* must be chosen

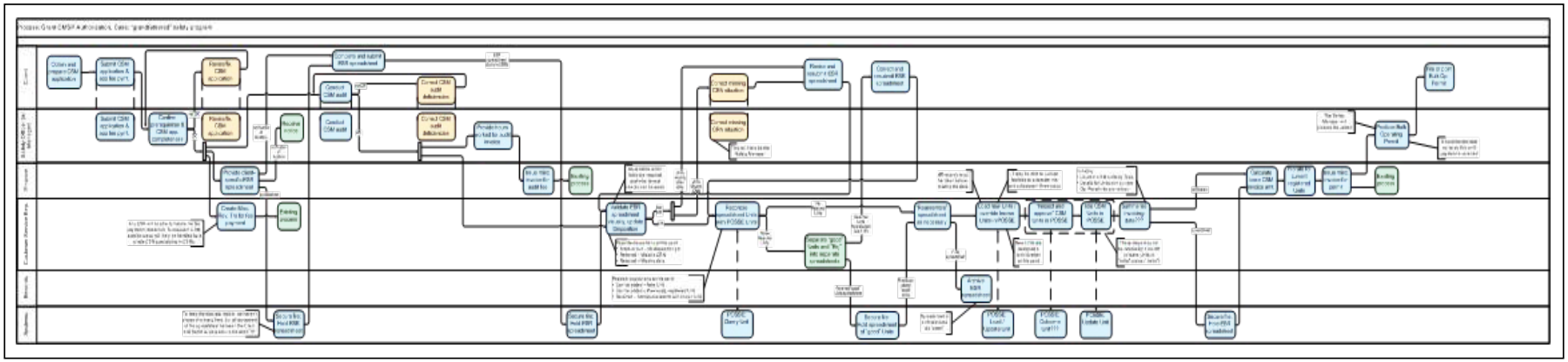
Enabler – A factor that is adjusted to impact process performance.



BOPDM: Business-Oriented Process & Data Modelling

We model the as-is process to support assessment by enabler

As-is modelling maps *reality* – *who*, does *what*, *when*.



After as-is modelling, assess the process by *each* enabler, *one at a time.*
This provides a *fact-based* assessment of the *as-is.*

Process Workflow Design: Is each step adding value, placed at the right point in the process, sequential or parallel as appropriate, performed by the best role, etc.?	***Information Systems & Technology:*** Are the process, the steps, and the actors supported by the right systems and technology?	***Motivation & Measurement:*** How is the performance of the steps, the actors, the participating functions, and the process measured, and what are the consequences?	***Human Resources & Organisation:*** Are roles suitably broad, are organisations designed properly, and are roles & skills deployed well into the process?	***Policies & Rules:*** What policies or rules , whether internal or external, constrain or are enforced by the process, and what is their impact?	***Facilities (or other):*** Are the layout & furnishings optimal or do they impede the process? (Many clients instead use this enabler to consider data, info, and knowledge.)

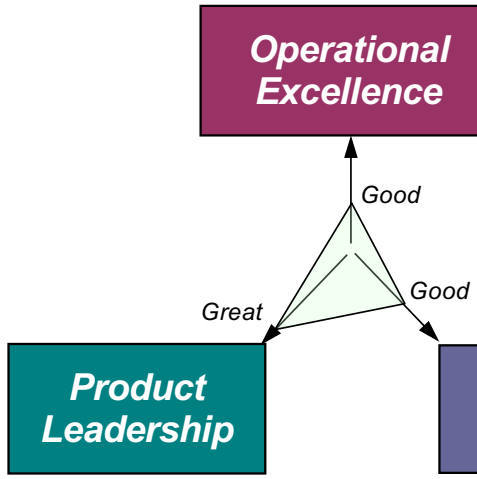
5. Process goals: know your "differentiator"

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. Enterprise system implementations must include a business process perspective
4. Success with business processes requires a *holistic view* in which six *enablers* are considered
5. A business process can't be great at everything – a single *differentiator* must be chosen

As noted, this is one of the things I do on ~100% of *Project Recovery* assignments -

1. Build *Process Scope Model* & *Process Summary Chart*
2. Develop *Case for Action* – an *As-Is Assessment by Stakeholder*
3. Establish the *Differentiator* ← *Why?*
4. *(*Optionally conduct an *As-Is Assessment by Enabler)*

Great processes don't try to be all things to all people –
they strive to be ***great*** at one differentiator, and ***good*** at the other two…



Consistent, predictable, error-free, and efficient – focus on low cost (or safety, in some industries.)

More efficient, but less flexible in changing direction or meeting needs of individual customers.

Continuous and rapid introduction of new products and services, or changes to the mix

More flexible for adapting to needs of new offerings, but less efficient.

Tailors product or service delivery to the processes of individual customers.

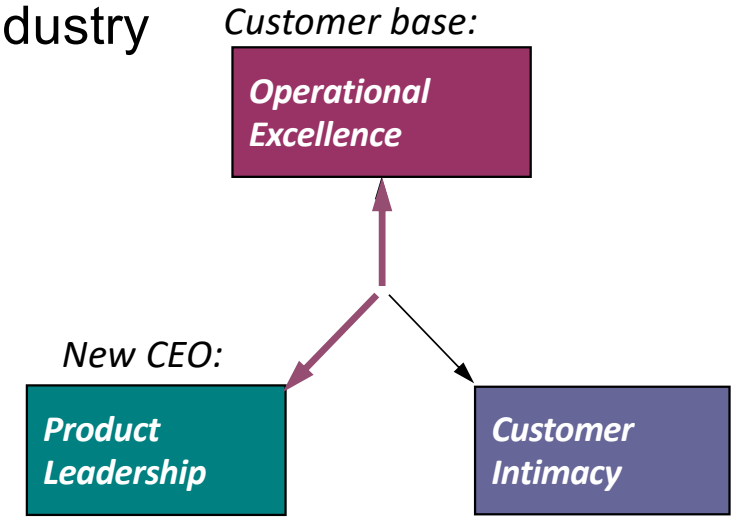
More flexible for adapting to needs of individual customers, but less efficient.

The original reference:
The Discipline of Market Leaders
Michael Treacy and Fred Wiersma
Addison-Wesley 1995

Example: "differentiator confusion"

Getting it wrong can be *expensive…*

- Insurance company recruits CEO from high tech industry
- New CEO decides "innovation is everything" – $100M spent on process redesign and system development in support of "innovative car insurance products" – *Product Leadership*
- Total failure – customers wanted affordable, easy to understand, easy to buy insurance – *Operational Excellence (Op Ex)*



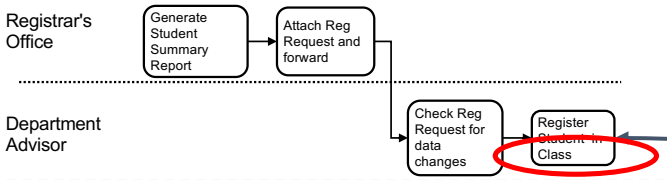
Key point! Everything relies on the Concept Model / Data Model

All use the language and constraints of the Concept Model (the "thing model") – the ultimate "what"

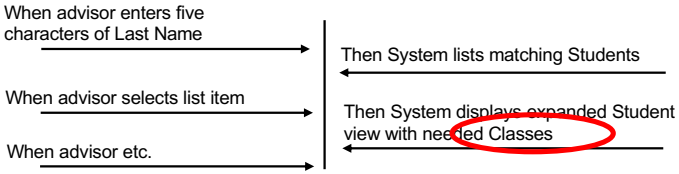
Business Goals & Objectives

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.

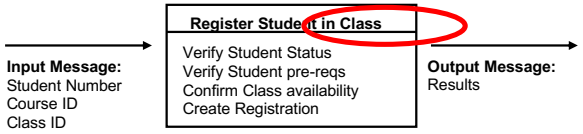
2 Business Process



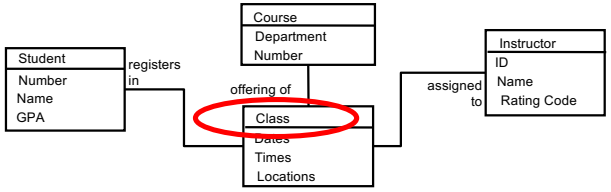
4 Presentation Layer (user interface)



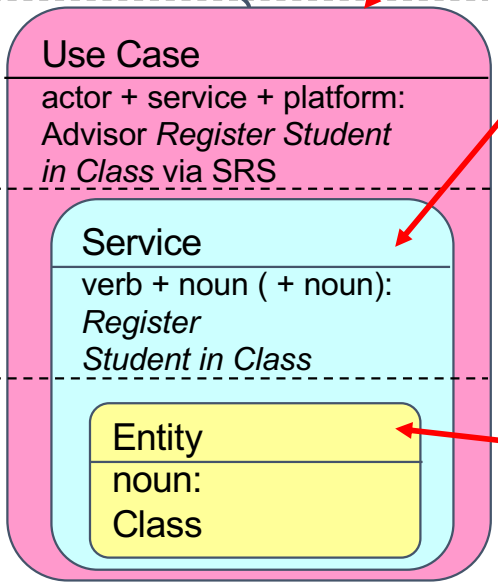
3 Application Layer (rules & logic)



1 Data Layer (data & storage)



My usual sequence



Use Cases/User Stories:
- Who (Actors) needs access to the Services, and how (Platform)?

Verb-Noun pairs:
- The *Services* (event-handlers) that are at the heart of a *Service Oriented Architecture.*
- *Also* "building blocks" of Business Processes

The core *Nouns or* Things in your enterprise. Also known as *Business Objects.*

Bonus – great starting point to discover your Events/Services and Use Cases/User Stories

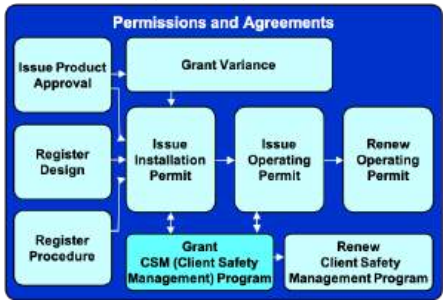
Another key point! Different levels of detail for different purposes

Different models and levels of detail for different audiences and purposes.

Also applies to Use Cases, Services, and Data Models

Scope – for Planning

Process Landscape (optional):



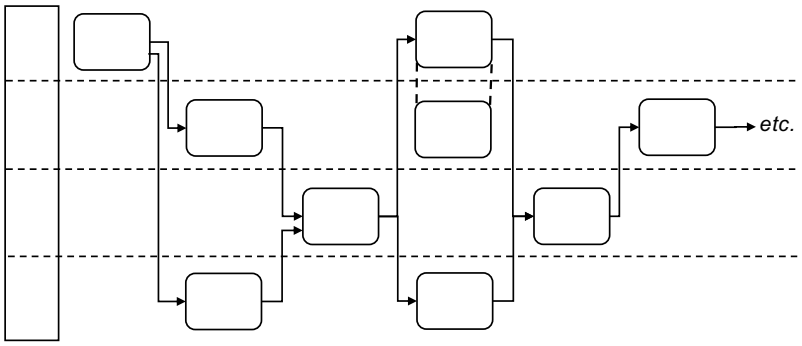
Process Scope Model:

Process Summary Chart:

Boxes

Concept – for Understanding

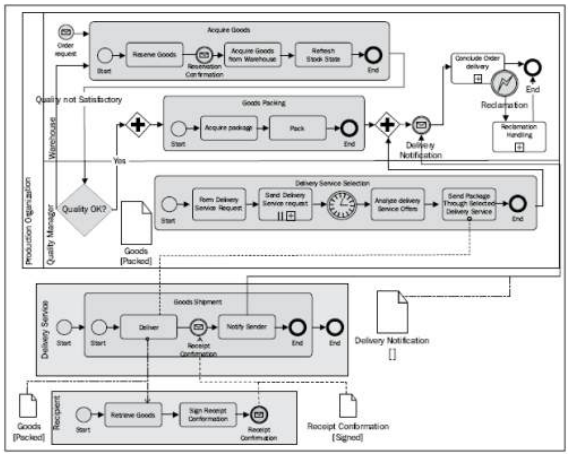
- Augmented Scope Model showing next level activities: *who - what - how*
- "Business-friendly" (just boxes & lines) flow models to maximise communication and participation
- Two levels – *Handoff* and *Service*



Boxes & Lines

Detail – for Specification

- Detail for technical design, perhaps using full BPMN



Boxes, Lines, & MANY Symbols

BOPDM: Business-Oriented Process & Data Modelling

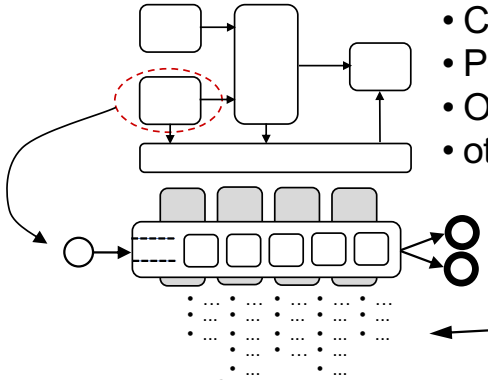
Our three-phase methodology – proven, practical, & agile

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***



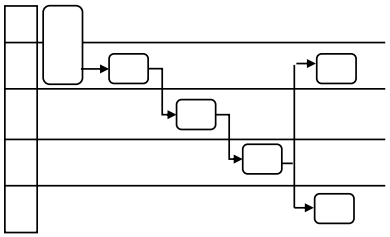
- Customer
- Performers
- Owner
- others…

- 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*

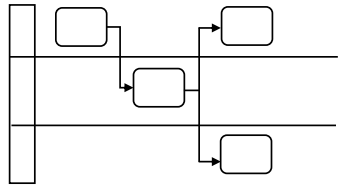
3 Design the To-Be Process

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

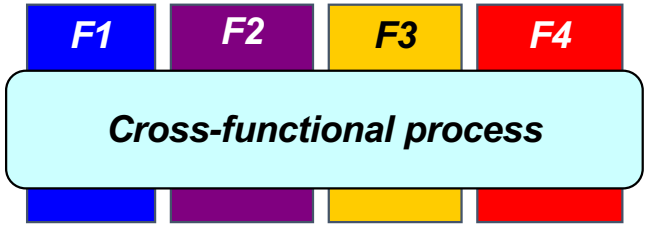
Design the to-be process:
1 - ***essential*** activities first
2 - "who & how" next
3 – transport & protocol last

- Select key to-be *Features*

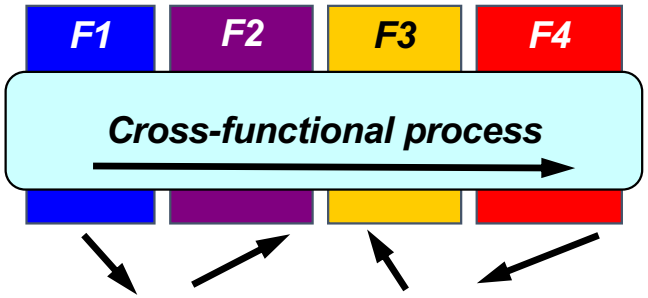


- Assess each key *Feature* by enabler
  - Identify and sequence *essential* activities
  - Develop *Workflow Models* for essential activities by adding *who* and *how*
  - *…on to requirements definition and implementation*

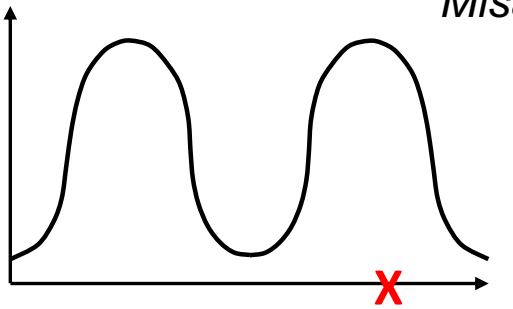
Five key points plus a BA framework plus a methodology



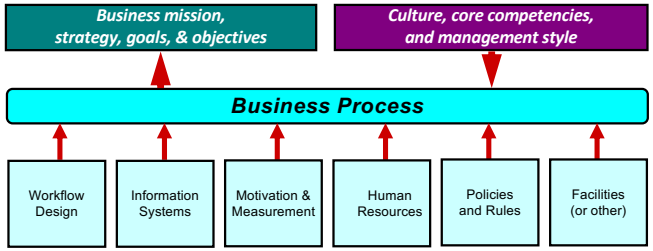
Processes:
"large" and X-functional



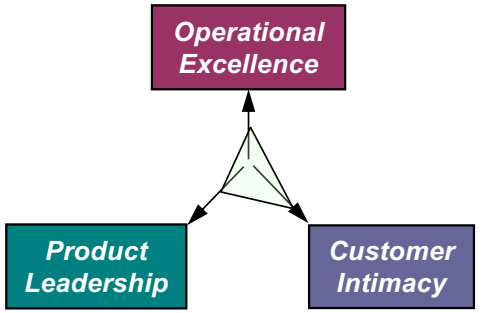
Misaligned measures



System success

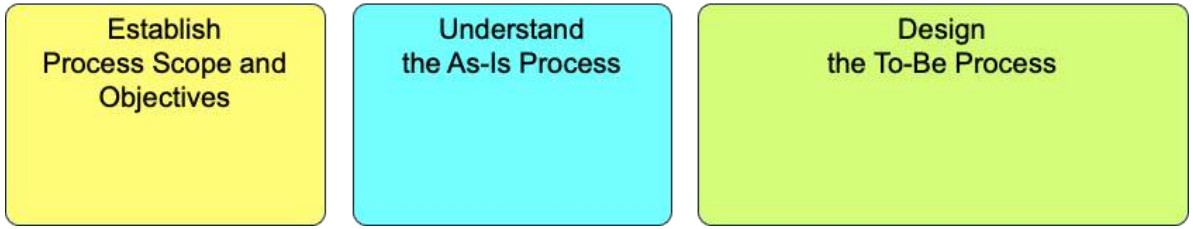


Holistic method



Differentiator

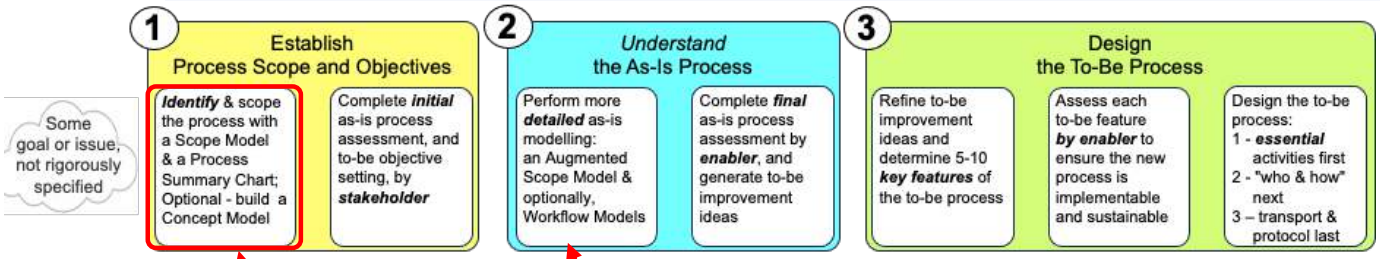
…and a proven Methodology



Model-driven framework

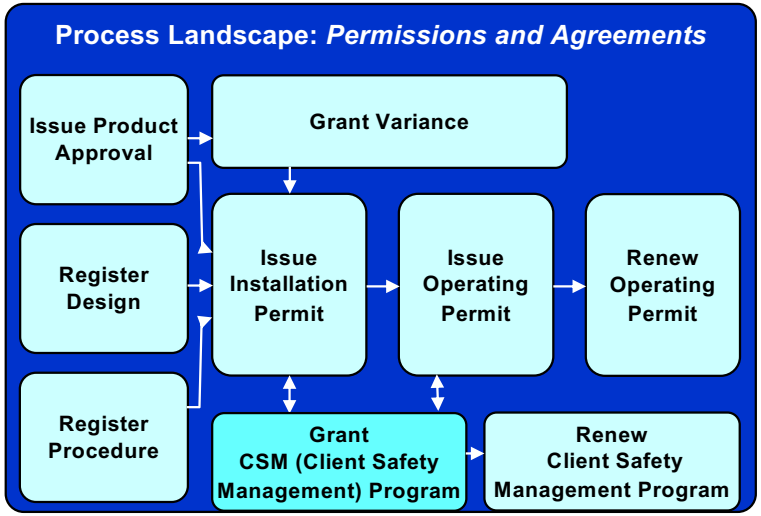
- Process Modelling
- Use Cases
- Service Specification
- Concept / Data Modelling

Identify & scope process(es)



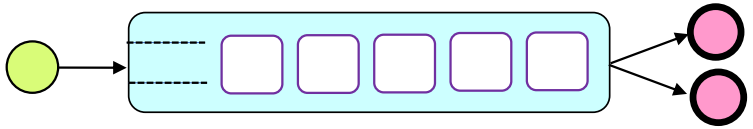
Clarify scope and context

I used to dive in here…
… lots of issues!

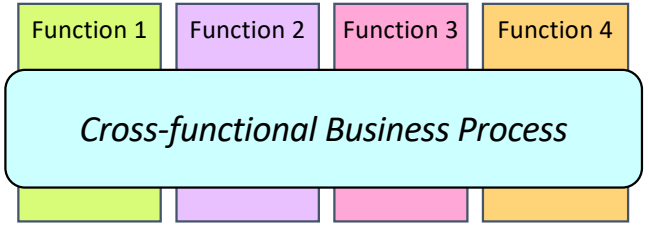


Whether it's a new initiative or "project recovery," ***always:***

- Develop a Process Scope Model



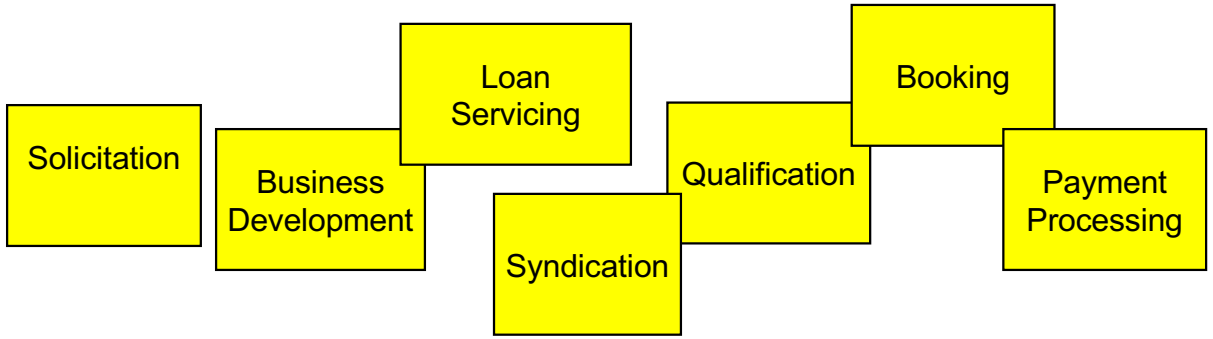
- Develop a Process Summary Chart



You *might* start at a higher level, with a ***Process Landscape*** –
a decomposition of a business area into a family of *individual business processes*

Process discovery example

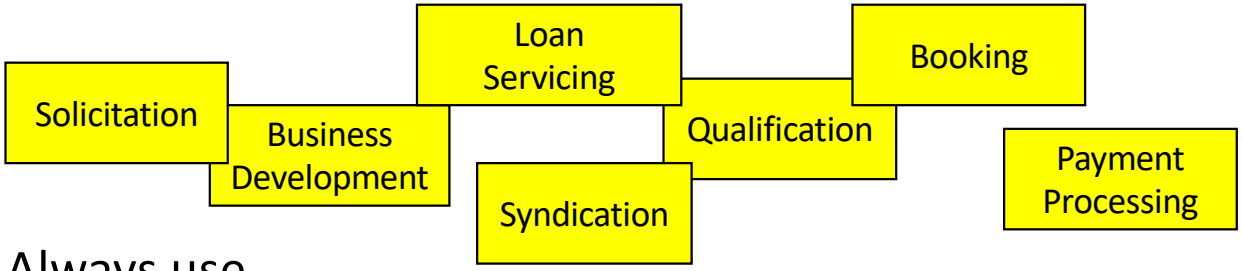
A bank believed they had identified the 12 *business processes* in their Commercial Loans Management area, including these 7:



Discuss:

- What is wrong with the names of these processes?
- Can you think of any questions to help improve these process names?

Bottom-up process discovery – example

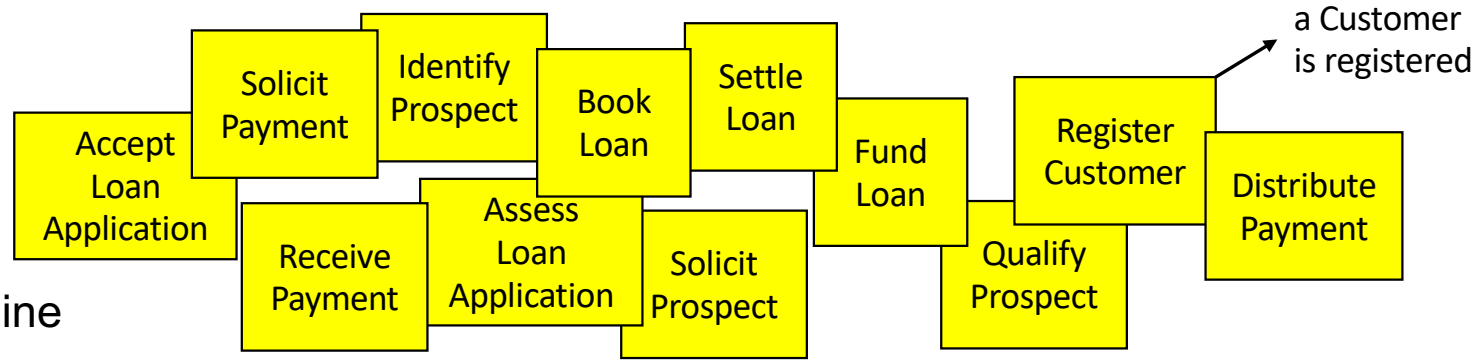


Dubious "business processes"

Always use
"active verb – noun" naming
with no "who and how"

Client then identified *recognisable* activities, each producing an essential *result* (easy!)

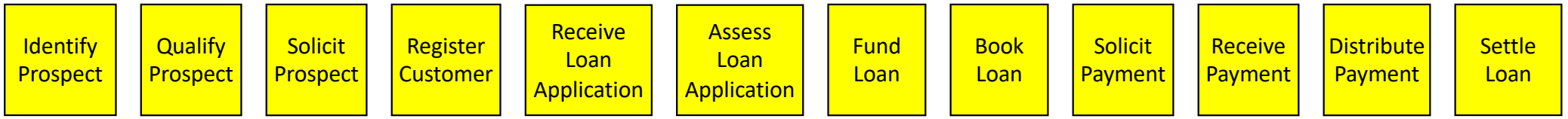
Let's put these
in sequence, then
use *TRAC* to determine
Business Processes.



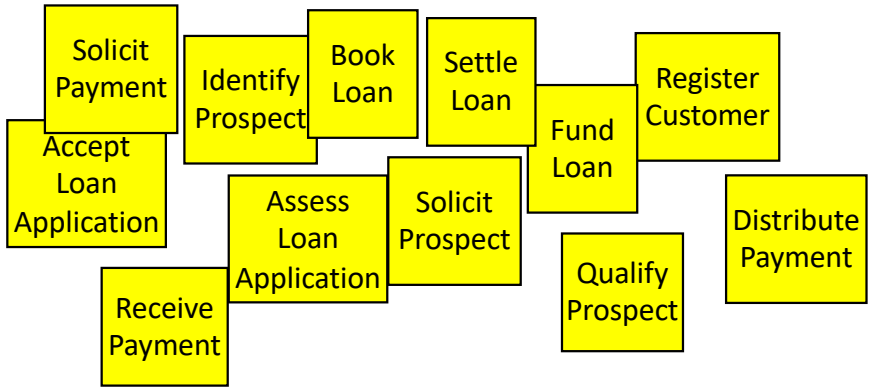
This was done in-person with Post-its and flipcharts
but tools like Lucidchart and Miro work well virtually

Summary – sequence activities

Not usually linear – parallel chains are typical



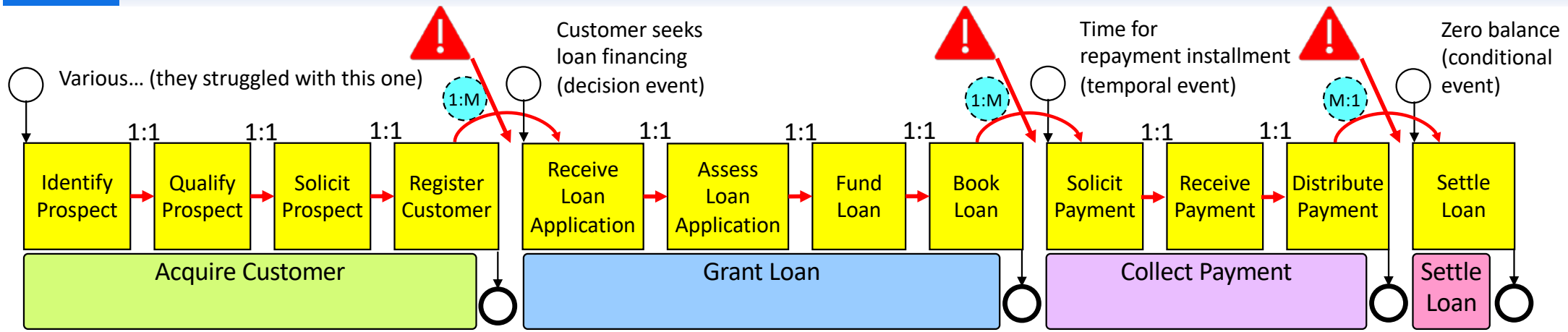
The clients arranged the activities in sequence:
- *easy!*
- *a learning experience!*



Now we'll use my "TRAC" framework for business processes –

- *Trigger*
- *Results*
- *Activities*
- *(Cases – later)*

Summary – use TRAC to discover business process boundaries



It appears we have discovered four business processes, each with:

- Trigger
- Results
- Activities

(Cases later)

Customer:
an Account that enables business with the bank
The Bank:
a new Customer (an asset)
Business Development:
Commission credit

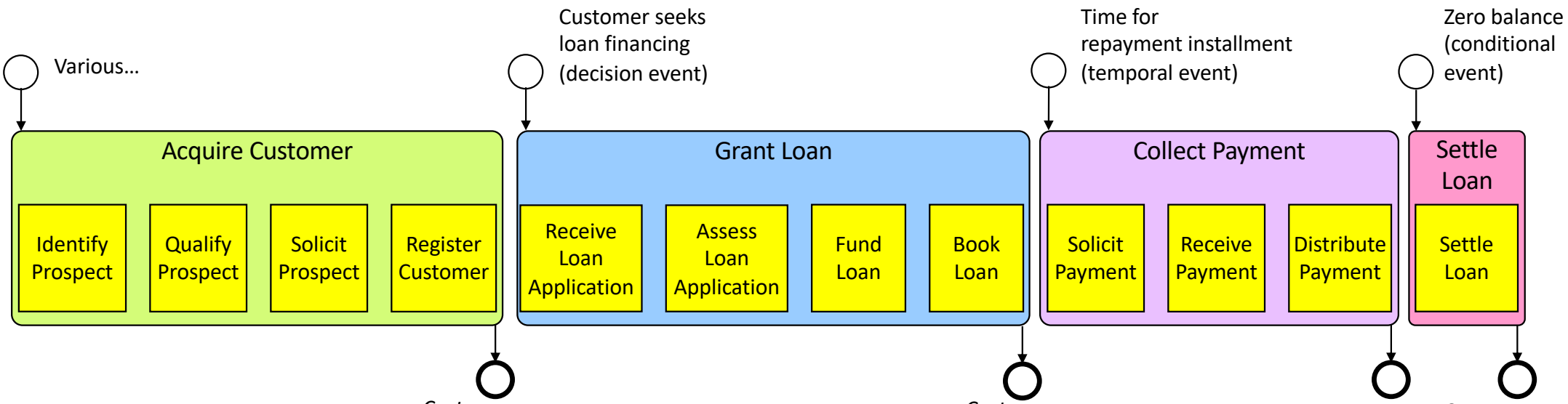
Customer:
Loan funds available
The Bank:
a performing asset (Loan)
Syndication Partners:
a share of the Loan

Bank:
Loan payment received & distributed
Syndication Partners:
Loan Payment received

Customer:
release of Loan liability
The Bank & Syndication Partners:
completed Loan
Regulator:
Loan completion notice

1. ID where a final Result of value is delivered to one or more (usually at least two) stakeholders – "happiness points"
2. Identify points where a Triggering event (decision, time, condition) beyond the organisation's control is required before activities can proceed
3. Identify "cardinality" of connections between Activities (1:1, 1:M, M:1)
4. Identify "tokens" flowing through the activities
5. Name business processes with active verbs and nouns (usually the tokens)

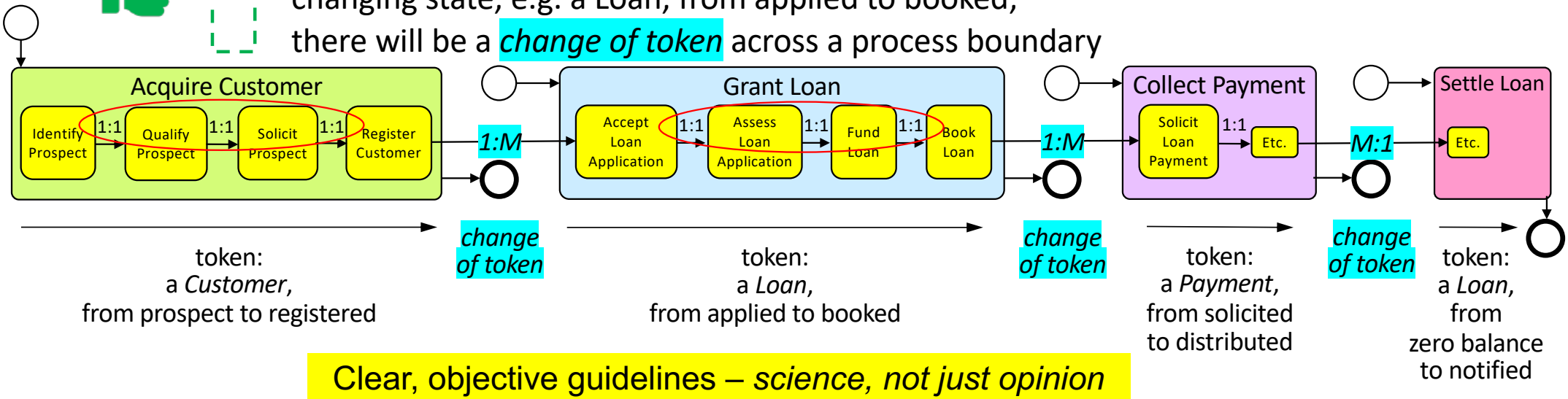
Four end-to-end business processes, objectively demonstrated



Client had faith these were *their* business processes

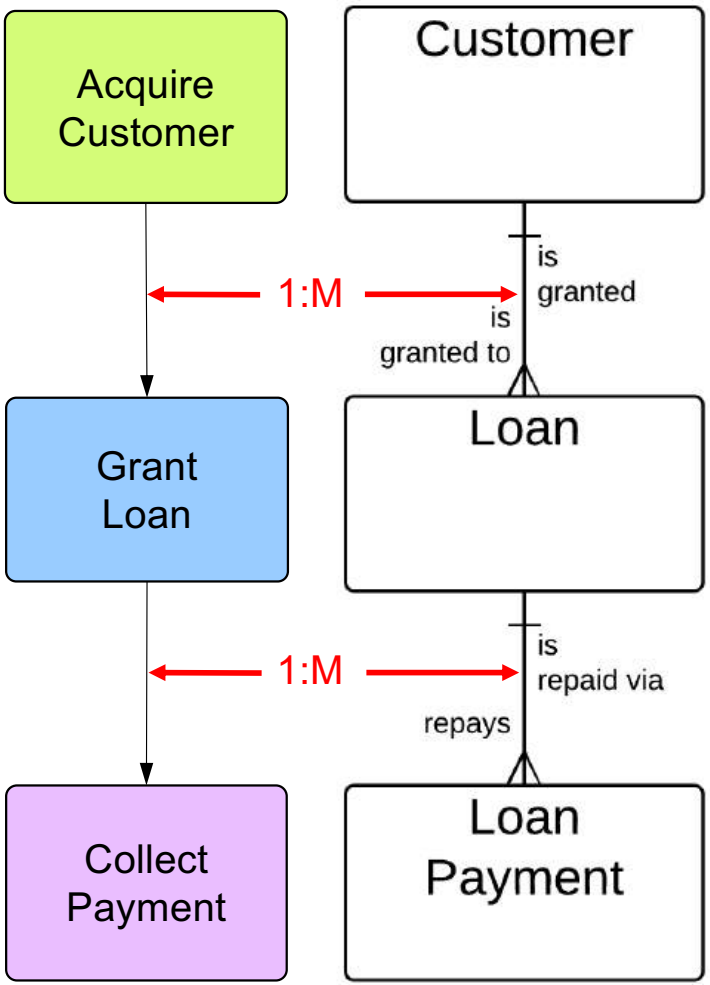
Six guidelines for well-formed processes, two clients really appreciate

1. "Active verb – noun" naming that indicates primary result
2. Triggered by an event (decision, time, data) outside process' control
3. At the end are results that makes one or more stakeholders happy
4. In between are ~5 to 7 major Activities (phases, milestones, subprocesses, …)
5. Activities linked *1:1* are probably part of the same process;
   a *1:M* or *M:1* connection between activities is probably a boundary
6. The same *token* moves through the whole process,
   changing state, e.g. a Loan, from applied to booked;
   there will be a *change of token* across a process boundary



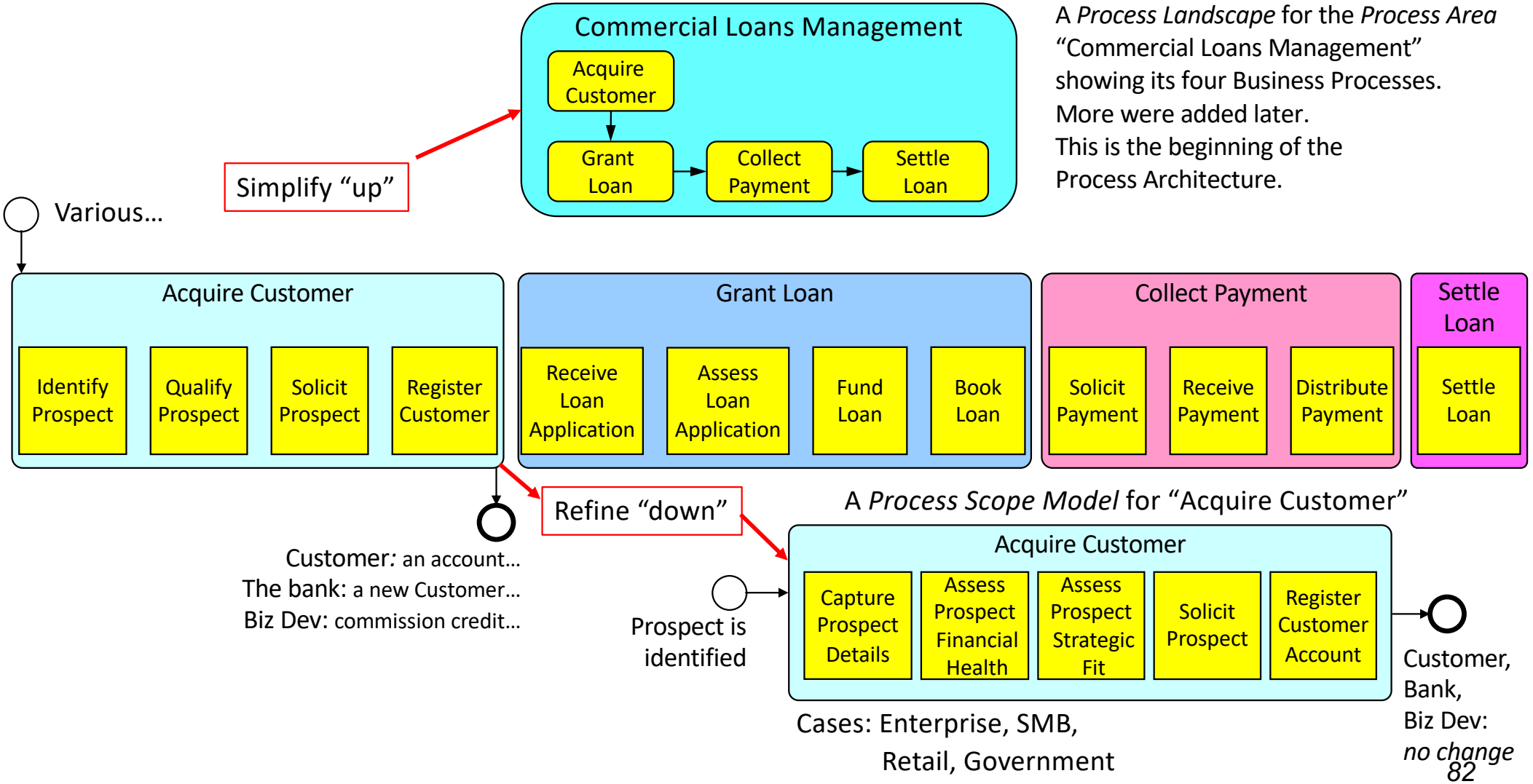
Clear, objective guidelines – *science, not just opinion*

Correspondence to the Concept Model



- The nouns in your verb-noun *Process* name are most often the *Entities* in your Concept Model, and each will usually have one primary *Process*
- The relative number of Process instances (e.g., 1:M or M:1) align with relationship cardinality
- This *does not* mean there is only one Process per Entity
  - Assess *Customer* Performance
  - Retire *Customer*
  - Merge *Loans*
  - Write Off *Loan*
  - …

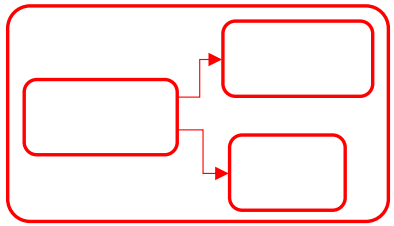
What next?



A *Process Landscape* for the *Process Area* "Commercial Loans Management" showing its four Business Processes. More were added later. This is the beginning of the Process Architecture.

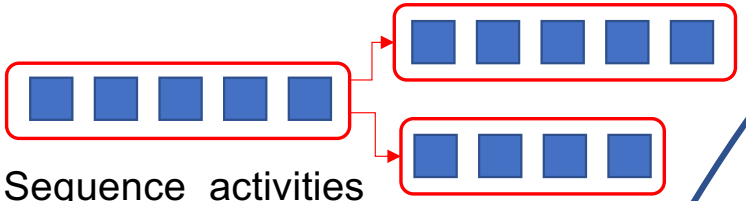
The arc of modelling and analysis

Start bottom-up to build overall framework – *Continue* top-down

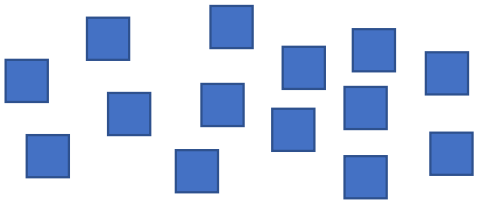


Draw a *Process Landscape,*
our overall framework

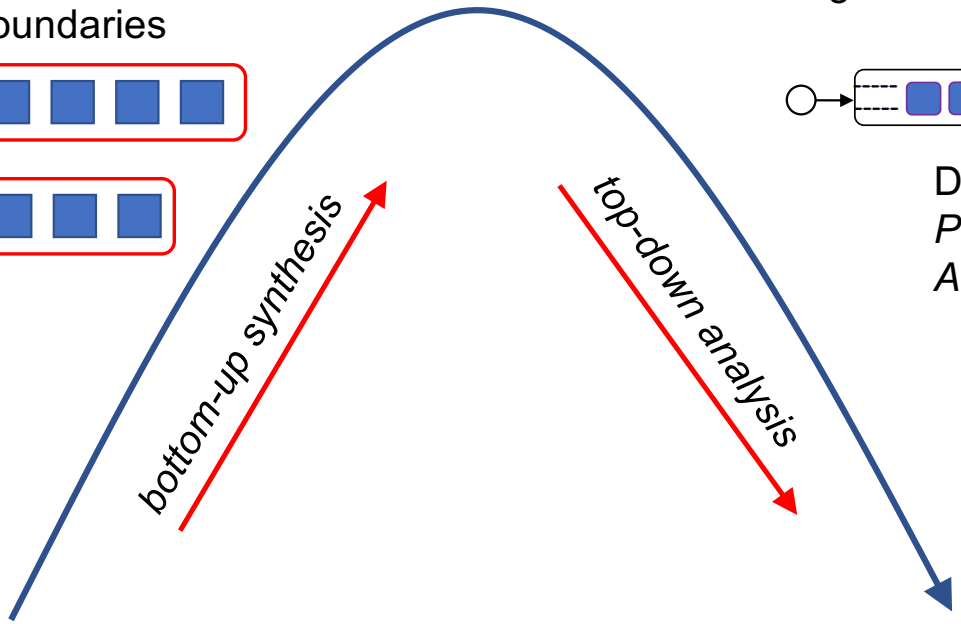
Use objective criteria to
identify process boundaries



Sequence activities
into “chains”

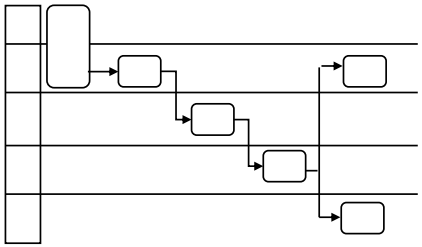


Granular activities
(real, recognisable work)



Progressive refinement and elaboration

Develop *Process Scope Model,*
Process Summary Chart, and
Augmented Scope Model



Develop *Initial Swimlane Diagram*

Our methodology – three responses to three common difficulties

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***

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

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

Big picture first

Don't start here!

Flow first, detail later

My *hardest* assignments

1 – Premature diagnosis of the situation

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

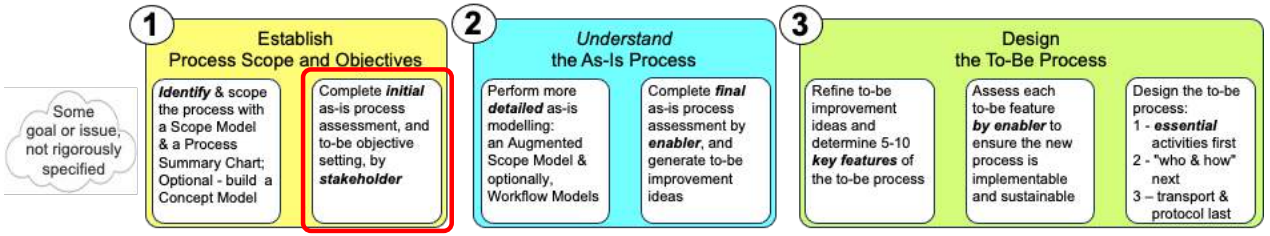
3 – A rapid descent into unhelpful detail

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.

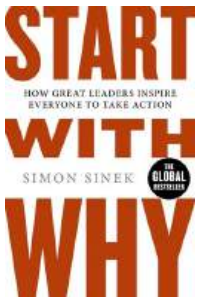
Clarify the big picture, then take a *controlled* descent with well-defined levels of detail.

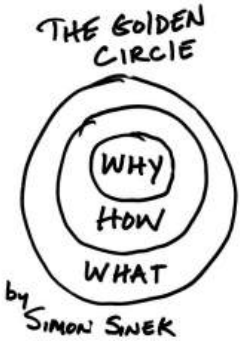
Perform initial as-is assessment, determine to-be objectives



Why does this process need to change?

But for a process…





"People don't buy *what* you do, they buy *why* you do it."

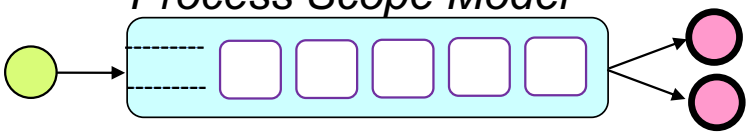
What first

Who & How next

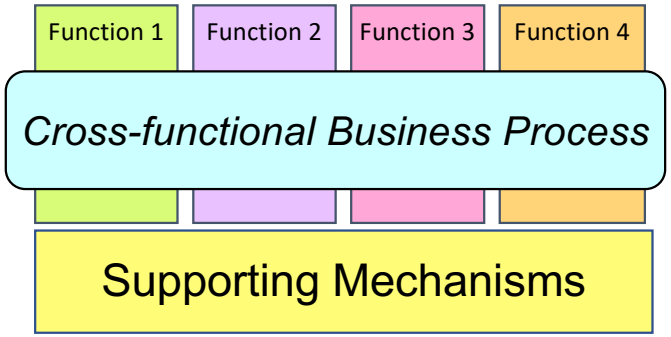
Only then *Why?*

Why does this process *need* to change?
We'll answer that with a *Case for Action*
(a nuanced form of problem statement) that encourages people to support *change.*

Process Scope Model



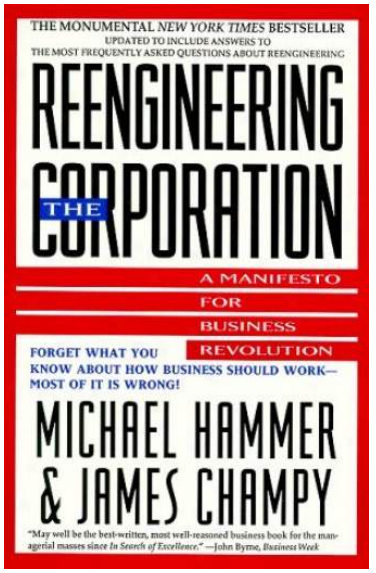
Process Summary Chart



Now we have an end-to-end, cross-functional perspective.

My version of Michael Hammer's "Case for Action"

Simplified, re-sequenced, more stakeholder-focussed



1) Stakeholder assessment – makes it *real*

What are the concerns of *each* stakeholder group?

- Customer
- *Performers*
- Owner/manager (the enterprise itself)
- Others (regulator, partners, …) as needed

We're not that bad!

2) Context – makes it *blame-free*

What changes in the environment since the process was "designed" have caused these issues to surface?

Yay – It's not our fault!

3) Consequences of inaction – makes it *compelling*

What will happen if the process is left as-is?

We'd better get on with it!

1. Stakeholder concerns

"You must communicate in a clear and compelling way why the process has to change by completing the initial assessment for the as-is process."

Initial assessment – 3 components

Stakeholder assessment – makes it *real*

- Customer
- Performers (***what's in it for me?***)
- Owner/manager (the enterprise itself)
- Others, as needed

Initial assessment – typical questions

Customer:

- Are there too many interactions?
- Are rules, requirements, protocol reasonable?
- Can *your* work be located within the process?
- Are *you* the process integrator – the human glue that connects the process steps?

Performer:

- What are your major sources of frustration?
- Do you have the necessary tools and support?
- Are there redundant steps or steps that serve no purpose?
- Are problems caused upstream? Does the workload vary wildly?
- What would you change if you could?
- *Is* there a documented process?

Owner/manager:

- Does the process use resources you would rather re-allocate?
- Is it a net contributor or a source of problems?
- Does the process constrain innovation, growth, or opportunities?
- Is it a source of customer or media *criticism?*

2. Context – assessing changes in the environment

Context – makes it blame-free

What changes in the environment since the process was first "designed" have caused these issues to surface?
Consider external and internal influences.

Areas to consider:

- Regulatory change
- Workforce changes (e.g., "recruiting and retaining" vs. "retiring")
- Emergent technology (e.g. AI, robotics, drones, BP Automation, SMAC ("Social, Mobile, Analytics, Cloud,") or current technology is EOL ("End Of Life"))
- Changing customer expectations
- Competition, especially new or emerging *Current "Top Five"*

- Economic conditions
- Change in business volume (growth or contraction)
- Change in business model (e.g., customised or standarised)
- Change in business ownership (public, private,) M&A, divestiture
- Change in business leadership / executives
- Change in government (post-election fallout)
- Changes in business operating locations
- Socio-political change
- Environmental ("green") concerns

And many others (see "PESTLE")

3. Consequences of inaction

Consequences of inaction – makes it compelling

What will happen if the process is left as-is, and the status quo is maintained?

For the individual:

- Unsatisfying work environment?
- Diminished opportunities?
- Reduced employment or *loss of employment?*

For the organisation:

- Reduced performance?
- Reduced stature or reputation?
- Withdrawal from the market?

"Case for Change" example

Situation:

- Manufacturing firm redesigns core *Financial Reporting* processes prior to COTS selection
- No progress! – Project has descended into "the blame game"

Stakeholder assessment –

- *Customer* – financial markets / fund managers cannot get the info they need for investment decisions
- *Performers* – Finance staff spend all their time on assembling "the numbers" with no time for value-added analysis
- *Owner/manager* – CFO is under constant pressure and criticism from the financial markets and other executives

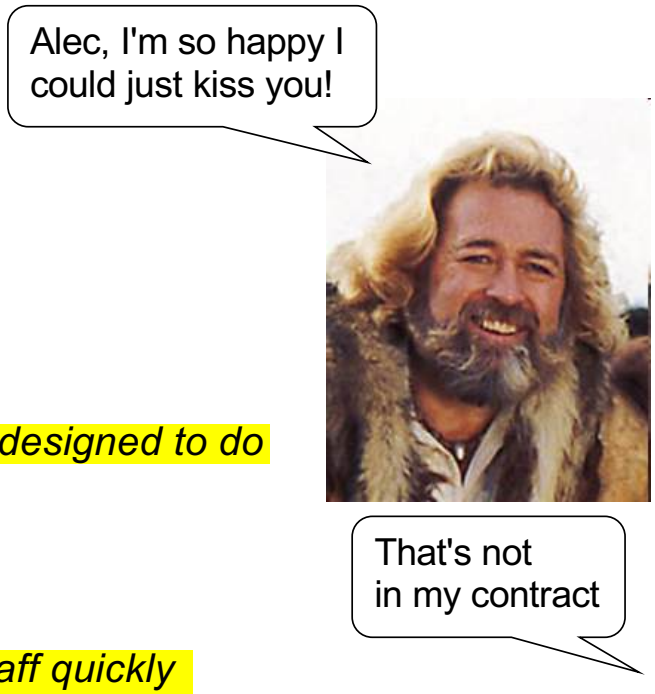
Context –

- Firm recently divested from a huge conglomerate
- Financial reporting was formerly to Head Office, but now is to financial markets which the processes were *never designed to do*

Consequences of inaction –

- Planned acquisition of competitor will not go ahead due to lack of financial market support for new bond issue;
- *Firm likely to be acquired by the competitor. Uh oh… Finance staff quickly realised their employment was threatened and got on board!*

Client was very happy!



Then, establish process goals / improvement targets

"You must also provide a sense of direction by defining to-be process goals and objectives."

Subjective goals

Give people a *"feel"* for direction:

- "Customers will love this process because…"
- "Performers will love this process because…"
- "The process owner will love this process because…"

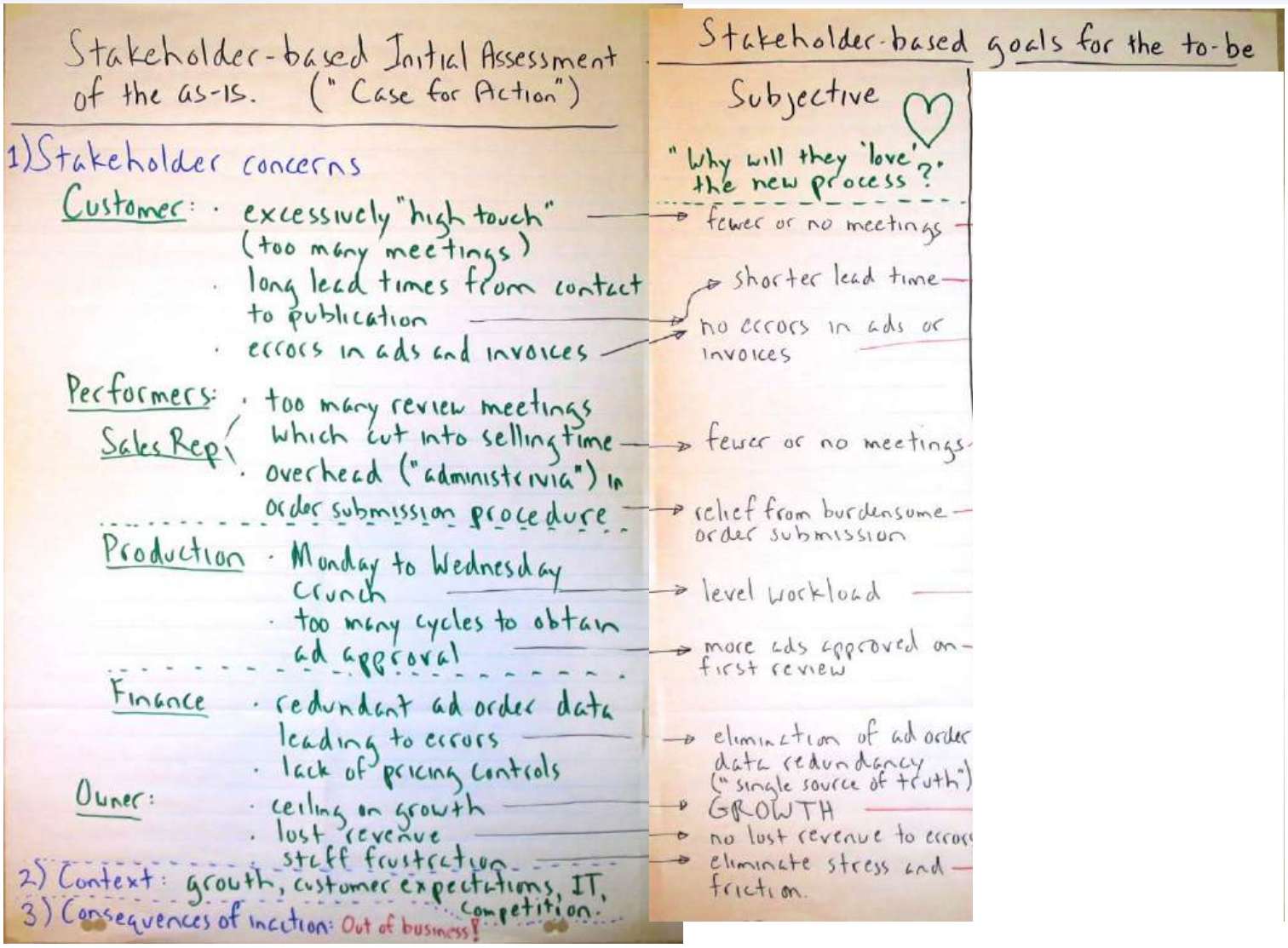
Measurable objectives

Provide *specific targets*
Establish baseline to *prove* success
Format:

- Topic
  (what will be improved?)
- Target
  (what is the measurable objective?)
- Timeframe
  (when will these results be realised?)

It may now be appropriate to introduce new process measures, metrics, and key performance indicators (KPIs) to establish baseline performance

Example from in-person workshop – assessment to goals



Concept Modelling Fundamentals

Course Topics

1. Requirements Definition
   - Goals, Issues, and the Return of Modelling
   - Case Study - Integrating the Techniques
2. Business Process Fundamentals
   - Five Things You Need to Know
   - Discovering, Scoping, & Assessing Your Processes
3. Concept Modelling Fundamentals
   - E, R, A - A Concept Model's Essential Components
   - Drawing Your Model for Maximum Understanding
4. Business Process Workflow Modelling & Design
   - Five Core Guidelines for Great Swimlane Diagrams
   - Facilitating a Process Mapping Session
   - Assessment of the As-Is and Transition to the To-Be
5. The Process-Data Connection
   - The Natural Synergy between Process & Data Models
   - Process-Data Synergies in Modelling, Analysis, & SW

The basics: ERA – *Entities*

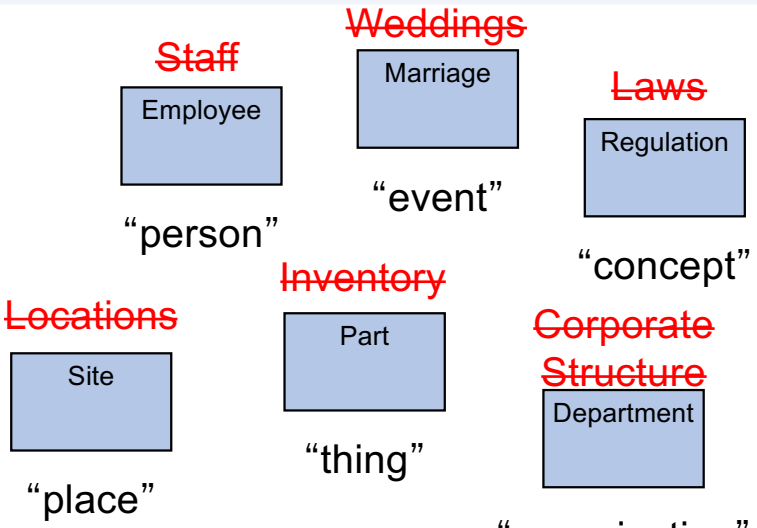
An *entity* is a distinct thing the business *needs* to know about - a *person, place, thing, event, concept,* or *organisation,* and…

- is named with a *singular noun* that implies a single instance
  - not a plural or collective noun, list, set, collection, report, etc.
  - we can discuss "one of them," e.g. "Weather" is not a good name
- has multiple occurrences (or instances)
- we *need* to and *can* keep track of (differentiate) each occurrence
- has *facts* that must be recorded, e.g.
  - *Student* attributes: Number, Name, Birth Date, Major, GPA, …
  - *Student* relationships: "majors in" *Subject*, "enrolls in" *Section*
- is acted on by *processes*, so they make sense in a "verb-noun" pair
- refers to the *essence*, not the implementation – *the most common error is to identify artifacts (forms, reports, spreadsheets, …) as entities!*

Named - a business-oriented noun / noun phrase

Defined - "What *is* one of these things?" or "What do you mean by _____?"

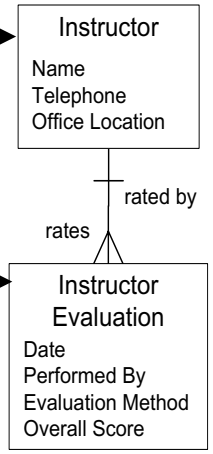
These are the ones our business partners often struggle with.



Two basic types

Independent
- can stand alone
- no relationships "on top" (no parents)

Dependent
- must have one or more parents – one or more relationships "on top" to parent(s)



Naming and definition – the essence of Concept Modelling

Agreement on naming is essential – entities are the *nouns* at the heart of business language. They are what processes act on, applications manipulate, databases record, and BI & analytics tools provide info about.

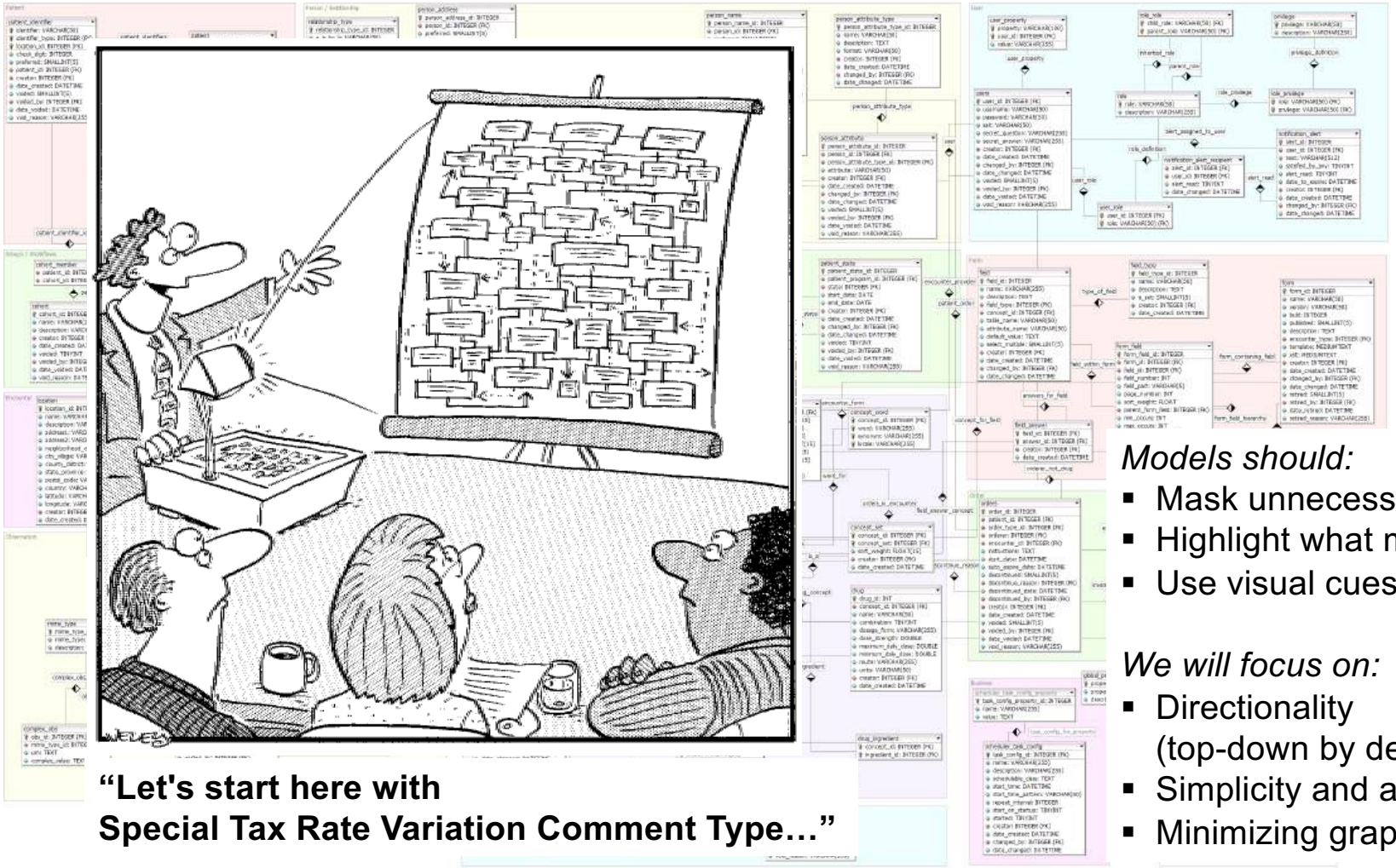
So, organisations need a *common language* more than ever, for…

- Data integration (data lake, data mesh, data fabric, data virtualisation, data warehouse, …)
- SLMs (*Small* Language Models vs. *Large*)
- Mergers/acquisitions/partnerships/…
- Business analysis – most requirements can't be stated without using a term from the Concept Model
- Performance measures, e.g., KPIs

Note – it works best if you don't start by talking about *Concept Modelling* or *Data Modelling*…

"Now! That should clear up a few things around here!"

Concept Modelling principles



**“Let's start here with
Special Tax Rate Variation Comment Type…”**

Models should:

- Mask unnecessary detail
- Highlight what matters
- Use visual cues consistently

We will focus on:

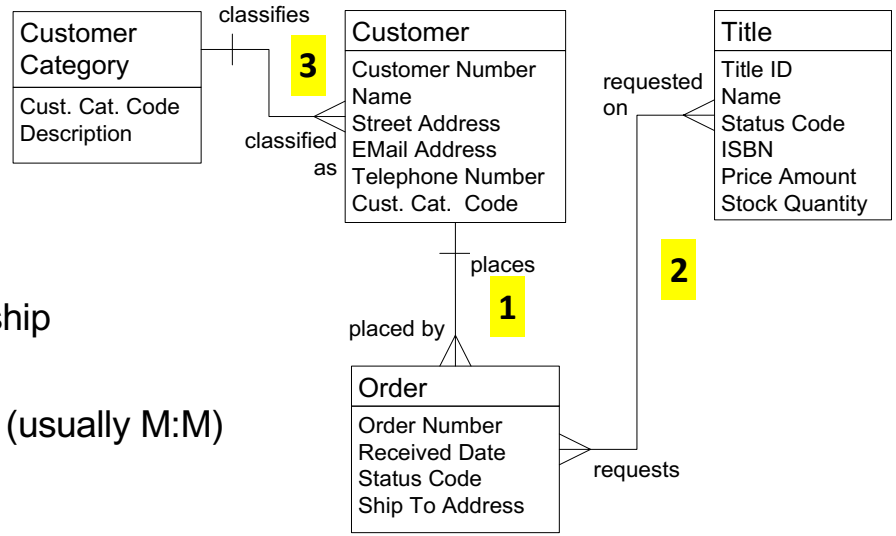
- Directionality
  (top-down by dependency)
- Simplicity and abstraction
- Minimizing graphic "widgets"

The basics – ERA – *Relationships*

An association between Entities that the business must keep track of

Named in both directions

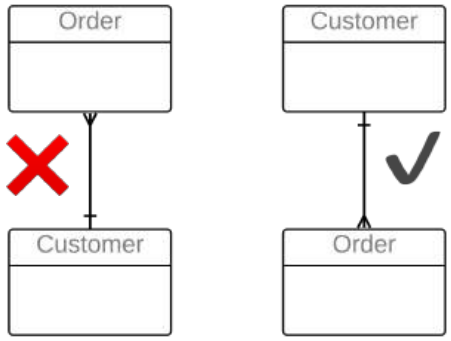
- verb-based phrase
- the line tells us they *are* related, the name tells us *how*



Different types of relationships

1. parent-child or characterising – "bottom to top" relationship from an entity to a dependent entity (1:M)
2. associating – "side to side" relationship between entities that are not dependent on one another (usually M:M)
3. classifying – "side to side" relationship from reference data to the classified entity (seldom shown in the Concept Model)

Dependency is shown top down – No Dead Crows

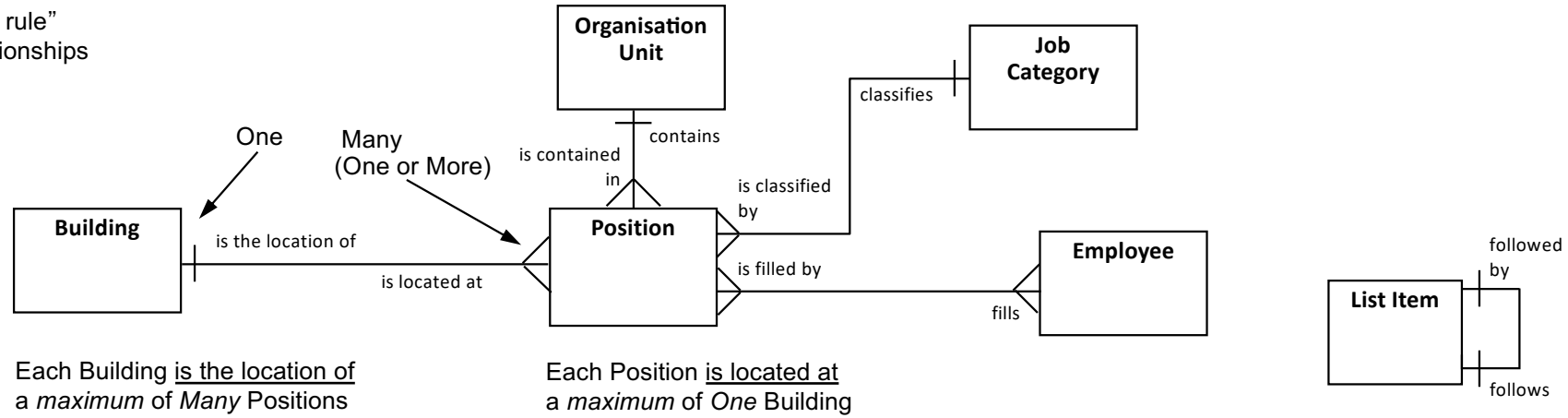


Relationships have rules

- cardinality – 1:1 (almost certainly wrong,) 1:M, M:M
- optionality – relationship *may be* present or *must be* present (not shown until later, in the logical model)

Relationship cardinality (maximum cardinality)

A kind of "business rule"
that applies to relationships



Each Building <u>is the location of</u>
a *maximum* of *Many* Positions
(or, ... *One or More* Positions)

Each Position <u>is located at</u>
a *maximum* of *One* Building

One to One (1:1) relationships in a conceptual or logical model are almost invariably an error except in recursive relationships.

To determine cardinality, *first name the relationships properly, and only then:*

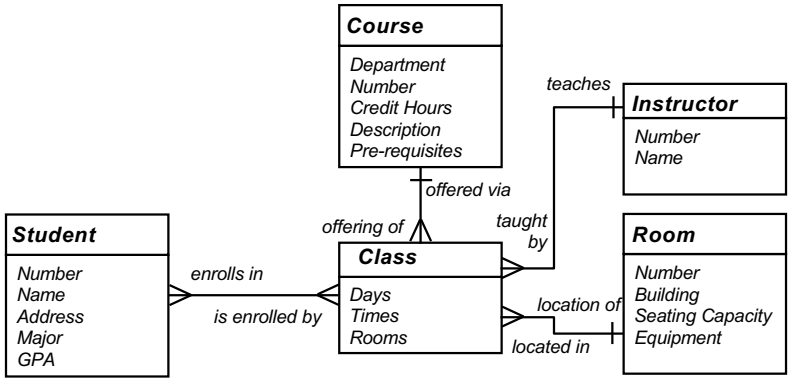
- for each entity, ask
  "Can one of these be related to a *maximum* of *One* of the other or a *maximum* of *Many* of the other?"

- record the answer (One or Many) at the "other" end;
  "One or More" works better for businesspersons than "Many"

- possibilities – 1:1 (error), 1:M (common), M:M (more work, eventually)

Relationships – state as assertions

1. You *must* state the relationship name as an assertion, in both directions (for clarity and confirmation)
2. Be clear on whether cardinality is “one” or “one or more” (don't worry about “may” and “must” at first)
3. *Emphatically* begin the assertion with the word “Each”
4. Try it on this model…



Note –
A Class is a scheduled offering of a Course during an Academic Time Period, e.g. a Semester or an Academic Year.
During an Academic Time Period there may be one or more Classes for a Course. Each Class is held on specific Days (e.g. Monday & Wednesday,) at specific Times (e.g. 10:30-11:30,) in specific Rooms (e.g. AQ3100 & CC7232.)

Each Instructor teaches one or more Classes (Sounds good…)

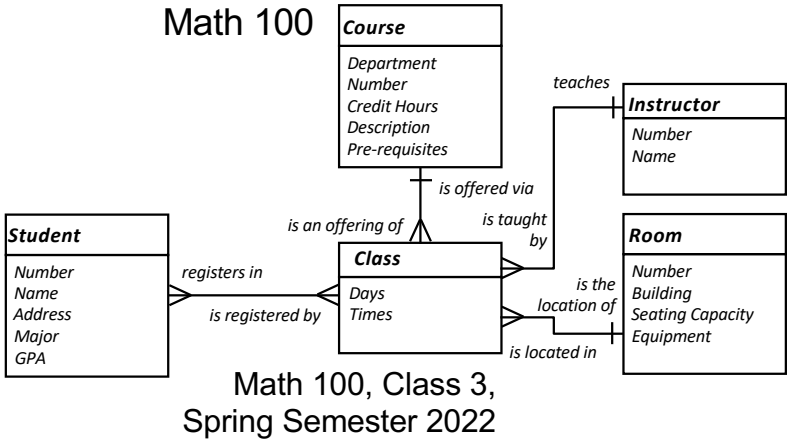
Each Class is taught by one Instructor…

1. Student-Class
2. Course-Class
3. Instructor-Class
4. Room-Class

Which ones might be *incorrect?*

Discussion – state as assertions, identify incorrect ones

In some universities, Students in the same Class could be earning credit for *different* Courses – it could be a M:M relationship.



1. Student-Class
   Each Student *registers in* one or more Classes
   Each Class *is registered by* one or more Students ✓
2. Course-Class
   Each Course *is offered via* one or more Classes
   Each Class *is an offering of* one Course ? – depends on Policy
3. Instructor-Class
   Each Instructor *teaches* one or more Classes
   Each Class *is taught by* ~~one~~ One or More Instructors
4. Room-Class
   Each Room *is the location of* one or more Classes
   Each Class *is located in* ~~one~~ One or More Rooms

Each Class is taught by One or More Instructors. On what basis?

- team teaching
- backup
- replacement
- specialist
- guest lecturer
- lab assistant
- teaching assistant
- …

We are discovering reference data to describe an Instructor's Role.

All of this has an impact on the Business Process! It's easier to resolve these rules before working on the Process.

The basics: ERA – *Attributes*

A fact about an entity recorded as a piece of data.
If facts are needed about a relationship,
we will later (in the Logical Data Model) create an entity that represents the relationship and records its facts

Like Entities, attributes are named and defined

Not every possible fact – just the ones we need

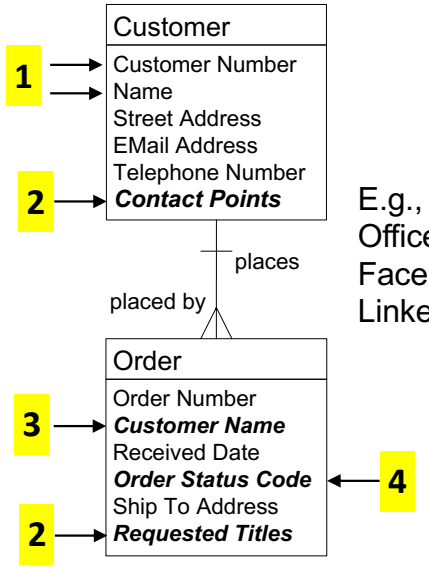
Have properties that we address during the transition from Concept Model to Logical Data Model

1. base or fundamental attribute
2. single-valued vs. multivalued –
   one attribute can have multiple values,
   at a time or *over* time
3. fundamental vs. redundant –
   the same value is recorded multiple times
   in different entities
4. "user-entered" vs. constrained –
   attribute can only come from a limited set,
   as in a drop-down list

Traditionally alphanumeric data; now includes richer types e.g., retinal scan image or voice audio clip

Eventually, in the logical model, an entity will contain only base / fundamental / *essential* attributes:

- an *essential fact* about that thing (entity)
- *not* multi-valued
- *not* redundant
  (a redundant attribute is an attribute that is really an essential fact about a *different* entity, so its value is recorded multiple times, redundantly)
- and *not* derived or calculated from other attributes; otherwise, clearly flagged "derived"



E.g., Mobile Number, Office Telephone Number. Facebook ID, LinkedIn ID, …

Summary – three types of data models

Different levels of detail support different perspectives

A more detailed version is on Page 136

1 Contextual (Scope)

- ✓ *Context model*
- ✓ Agreement on "big picture," context, and some vocabulary
- ✓ A block diagram of "subject areas," higher level than individual entities
- ✓ Shows the scope or "footprint"
- ✓ Optional – not useful on smaller projects

My most plagiarised diagram ever!

2 Conceptual (Overview)

- ✓ *Concept Model*
- ✓ Agreements on basic concepts, vocabulary, and rules

3 Logical (Detail)

- ✓ *Logical Data Model*
- ✓ Complete detail for physical design

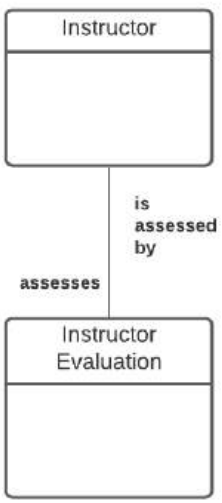
Some important differences

- ✓ Main ("recognisable") entities only - a singular noun used daily
- ✓ Main attributes only, many are non-atomic
- ✓ M:M relationships
- ✓ Doesn't show keys
- ✓ Not normalised
- ✓ A "one-pager"

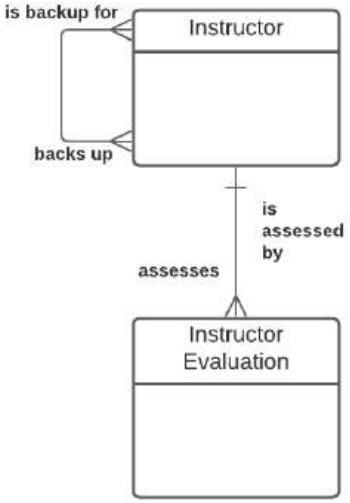
- ✓ All granular entities – many too detailed to come up daily
- ✓ All attributes included, all are atomic
- ✓ All M:M resolved
- ✓ Shows primary & foreign keys
- ✓ Fully normalised
- ✓ Five times as many entities

For reference – the Information Engineering symbol set

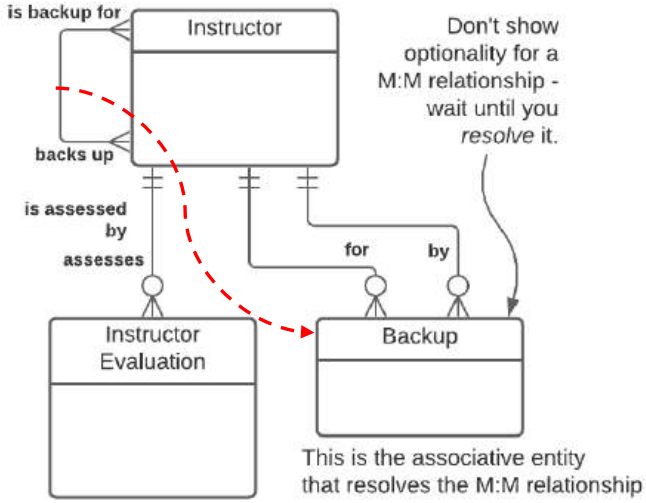
- This symbol set was refined and developed by Clive Finkelstein.
- Known in some tools as the "Martin IE" symbol set.
- Strengths are:
  - symbols are not "overloaded" – they explicitly convey only *one* idea.
  - can show as much or as little as needed in terms of rules.



The two entities are related - that's all this shows

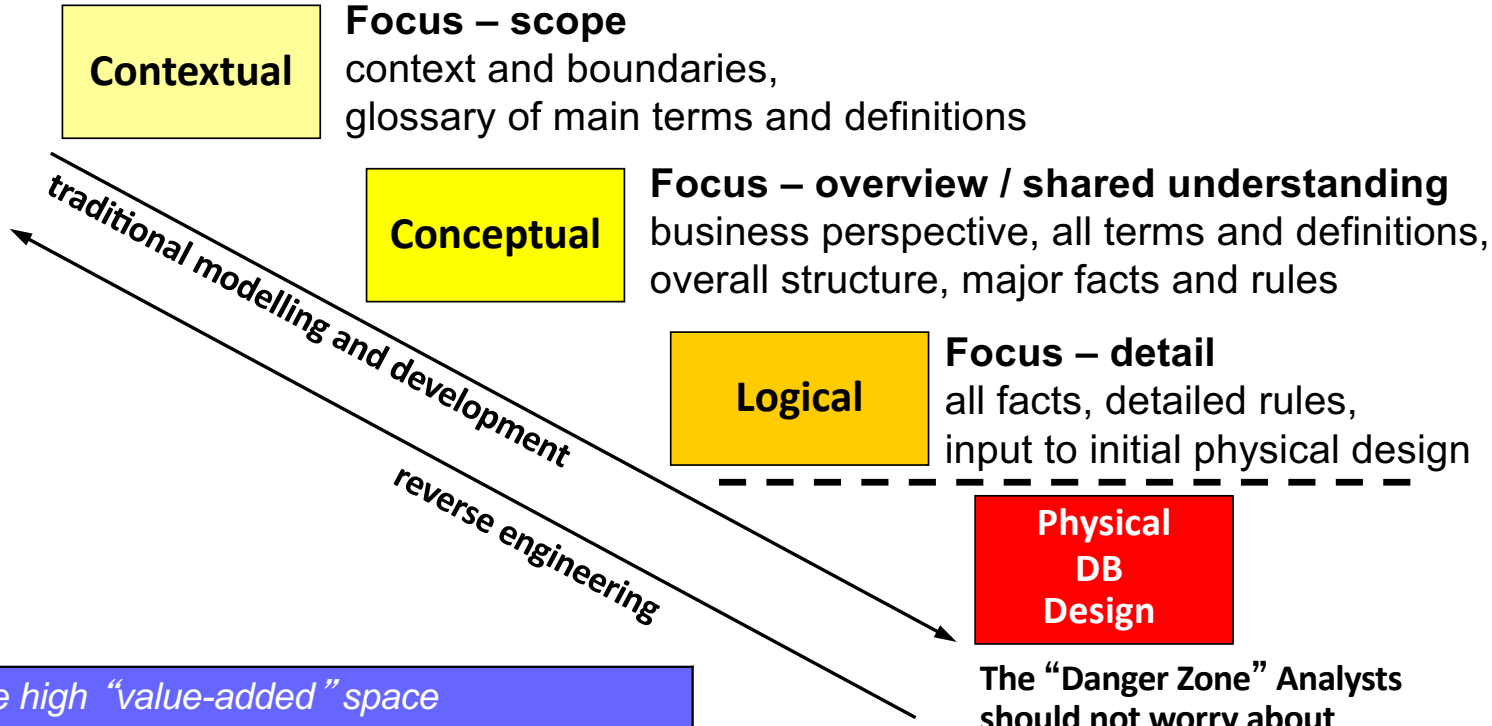


There is a 1:M relationship from the parent entity (business object) to the child entity (business object.) Optionality is not shown.



There is a 1:M relationship from parent to child, *optional* for the parent and *mandatory* for the child. (The parent *may* have a child, the child *must* have a parent.) This is by far the most common relationship in a logical model.

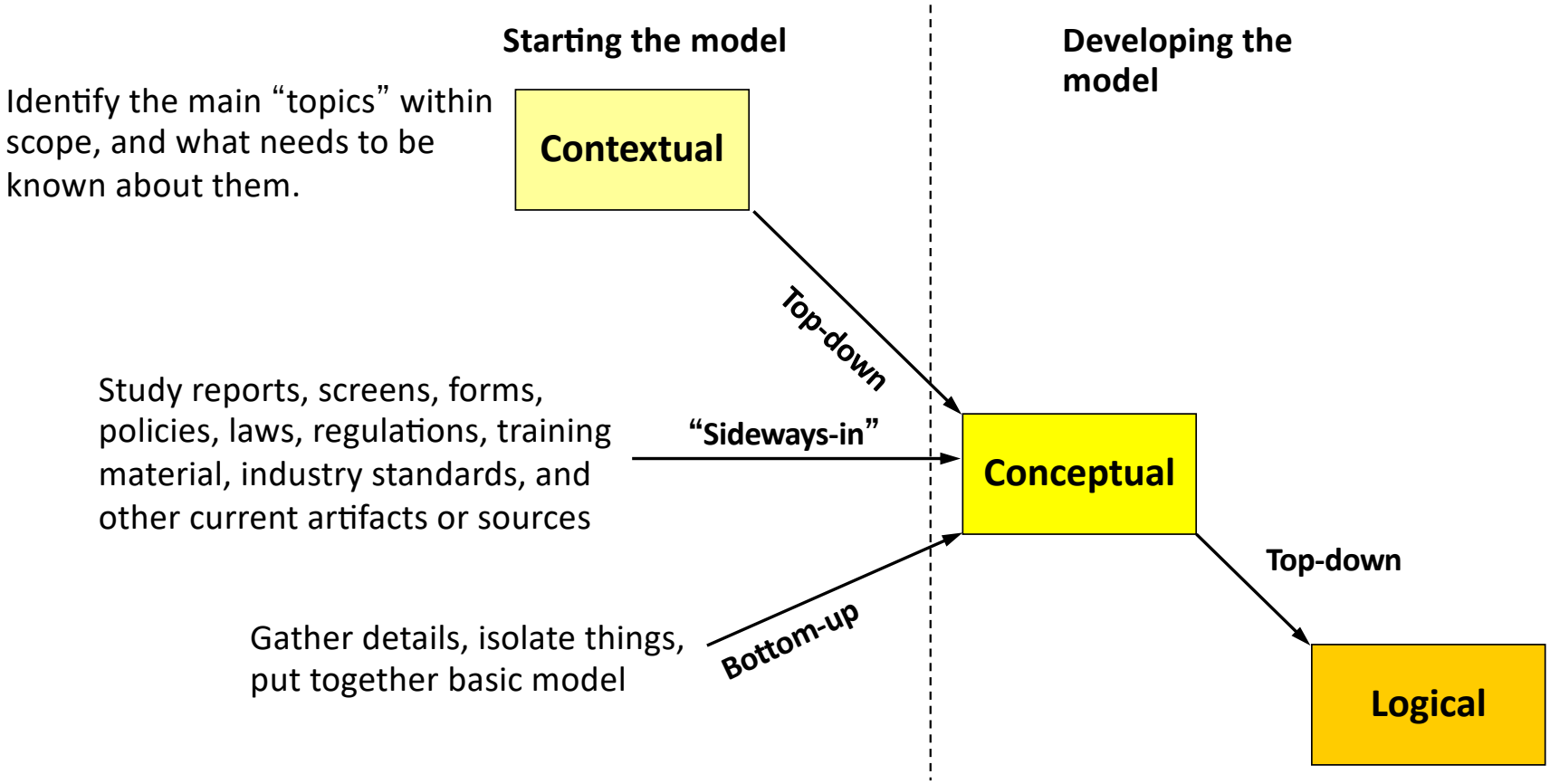
A natural progression



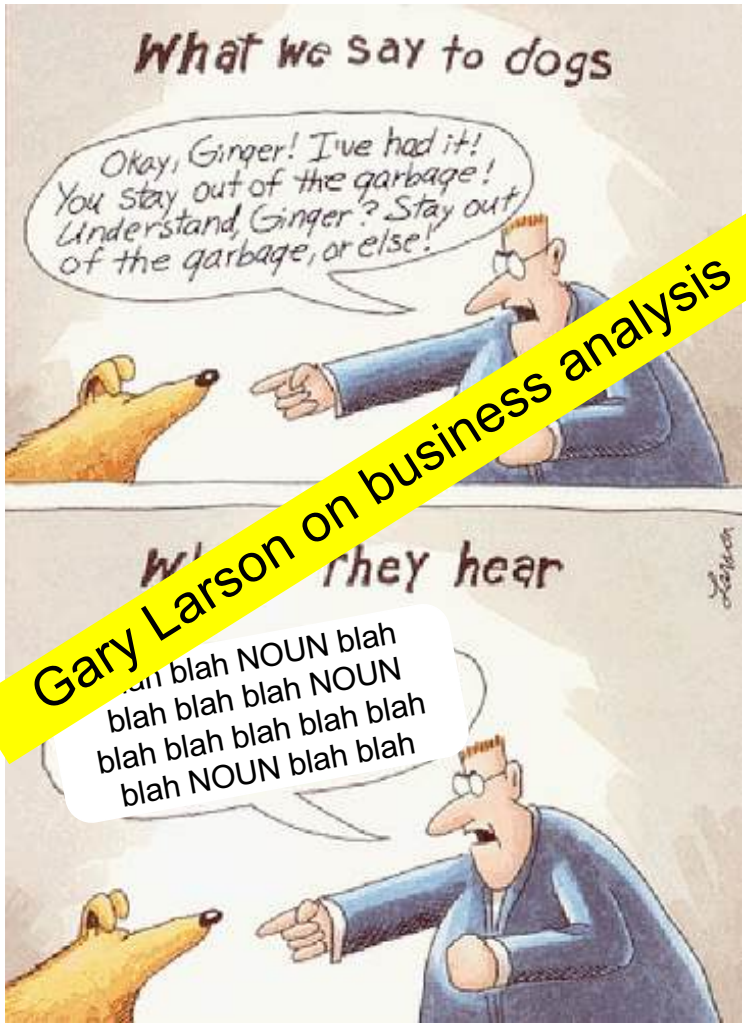
! *Get into the high "value-added" space*

- Contextual – helpful for large models
- Conceptual – a great way to add value
  - Improve communication among all players
  - Highlight disconnects – terms, rules, scope, …

Different ways to get started



Some advice on starting the concept model



<u>Don't</u> begin with a lecture on data modelling
(but I have a painful story that had a happy ending)

If you can, don't even mention "data modelling"

We use "terminology analysis" – starting with the <u>nouns</u> – at the outset of every project.
This was demonstrated earlier in the Client Safety Management example.

Starting a Concept Model bottom-up

1) Interview business representatives about their area:
   mandate and activities, goals and objectives, issues and opportunities,
   needs and wants, likes and dislikes, etc....

   Nod sympathetically but ignore it all (almost!)

   Instead, capture "terms" – anything that goes by a name.

2) Later, write each term on a large Post-it

3) In a facilitated session, participants sort terms into categories:

- Things (entities, but don't use the term… yet)
- Facts about things (add new "thing" if it's not there already)
- "Other stuff"

As needed, introduce criteria to be a"thing" (an entity)

"Other stuff" includes:

- Metrics
- Organisations, departments, jobs, roles, …
- Processes, functions, activities, tasks, …
- Systems, tools, equipment, mechanisms, …
- Reports, forms, screens, queries, …
- Other – too vague, only one instance,
  a "fact of life," not a thing we track, etc.

Exercise 1: Starting a conceptual data model

The assignment:
The following describes project tracking at Amalgamated Automaton. Read it over and be prepared to discuss the things about which the business needs to record information, and the important facts about them. The instructor will lead the development of an initial data model.

Amalgamated Automaton, Inc. has a growing Information Systems department. Until recent years, the department was concerned almost entirely with selecting, installing and maintaining purchased software packages. Recently, however, the focus has shifted towards the in-house development of application software.

One of the problems confronting the IS department is that they have no base of historical data to aid in trend analysis or estimating development effort, nor any effective means of charging back development costs. The proposed solution is to develop a simple Project Tracking System, which will work in conjunction with the existing Personnel and General Ledger Systems.

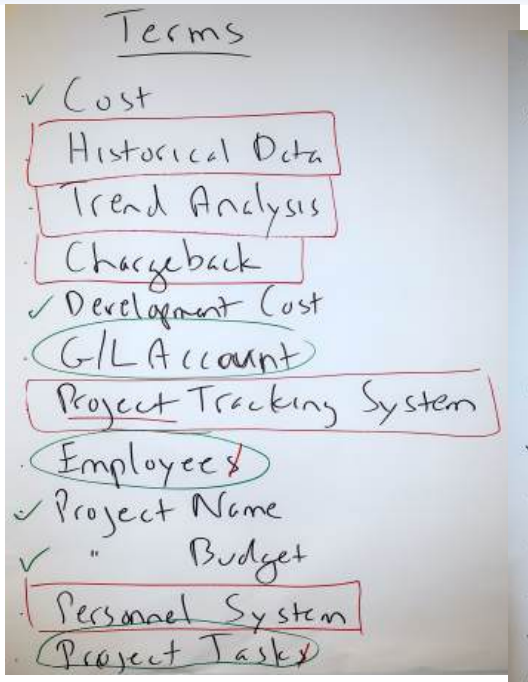
When a development project is initiated, a project name and a short description are recorded, among other things. Soon, before any further work is done on the project, a new account is created on the G/L System, identified by a G/L account number. Project costs will be charged to this account, and the project budget is recorded as the initial account balance in dollars.

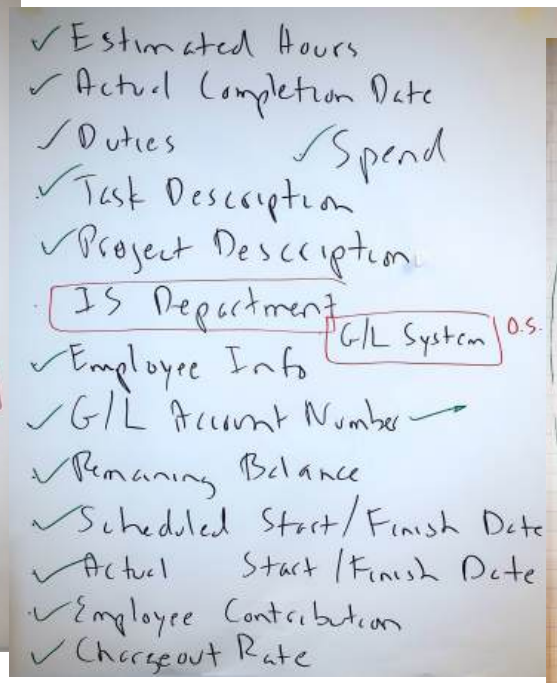
Project planners break a project down into many tasks, perhaps hundreds. A typical project task might be “Test Order Entry Module”. Some of the facts which are required about tasks include a brief task description, estimated work hours, and the scheduled start and finish dates.

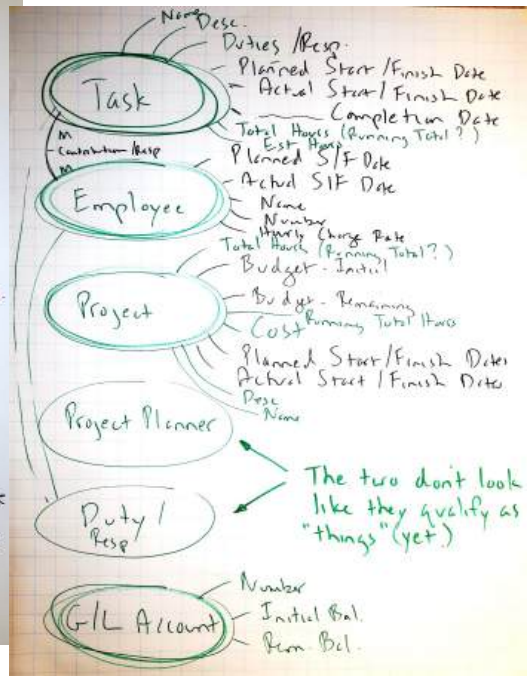
Eventually, individual employees are assigned responsibility for the tasks. Some tasks will be the responsibility of many employees, and an employee might be assigned to many tasks. As each employee is assigned to a project task, their planned start and finish dates, their contribution to the task (not a “kind of work,” but their specific duties on the task – e.g., “Develop test scripts”), and the estimated number of hours they are to spend on the task are recorded. Employee information such as the employee name and number are available from the existing Personnel System, although it will have to be modified to record the employee's hourly charge out rate.

When an IS employee begins work on a new task, their actual start date is recorded. A running total of the number of hours that they have worked on each started task is updated regularly. At the same time, the remaining balance in the project account is updated. When an employee completes a task assignment, the actual completion date is recorded.

Workshop example







Introduce "thing criteria" as necessary:

- *singular noun* – can talk about *one of them* (*Worker* not *Staff*, *Item* not *Inventory*)
- *multiple* instances
- must *need to* and be *able to* track *each* instance (uniquely identify each)
- has *facts* that must be recorded
- makes sense in a *"verb-noun"* pair
- *NOT an artifact* like a spreadsheet or report (not a Call Log or Worker Directory or…)

Identifying Entities – three common errors

1. Treating an "artifact" (a spreadsheet, report, web page, form, etc.) as an Entity – an Entity is a fundamental, singular thing with no reference to *implementation* – Artifacts reflect implementation (form, DB, spreadsheet, list, …) and contain attributes from
   - *multiple different* Entities or
   - *multiple instances of the same* Entity
   e.g., *"Admission Request Form"* or *"Orders Summary Spreadsheet"* or *"Daily Call Log" or "Materials List Fax"* or *"Class Roster"* or *"Course List"* or…

2. Identifying an Entity that exists in the real world, but whose *instances* can't be uniquely identified
   e.g., *"Transit System Passenger" or "Event Attendee"*

3. The "types vs. instances" problem – failing to clarify if the Entity deals with *types* of things (or *categories* or *kinds* or *classes* of things) vs. specific *instances* of things
   e.g., "*Test*" – is this a *type* of Test, or a *specific instance* of a Test?
   more examples coming…

Types vs. Instances example

Specifications for a type of vehicle – a 2011 Volkswagen GTI

5dr HB DSG

$31,275 MSRP

- ✓ Gasoline
- ✓ 6.3L - 8.7L fuel / 100 km
- ✓ Front-Wheel Drive
- ✓ 200 horsepower

Mechanical

Engine	Turbocharged Gas I4
Displacement	2.0L/121
Fuel System	Electronic Fuel Injection
Horsepower	200 hp @ 5100-6000 rpm
Torque	207 ft-lb @ 1800-5000 rpm
Steering	Power Steering
Drivetrain	Front-Wheel Drive
Transmission	6-speed DSG transmission

Interior

Passenger Capacity	5
Front Head Room	998 mm
Front Leg Room	1,046 mm
Front Shoulder Room	1,384 mm
Rear Head Room	978 mm
Rear Leg Room	902 mm
Rear Shoulder Room	1,361 mm

A "type" entity – Vehicle Make/Model

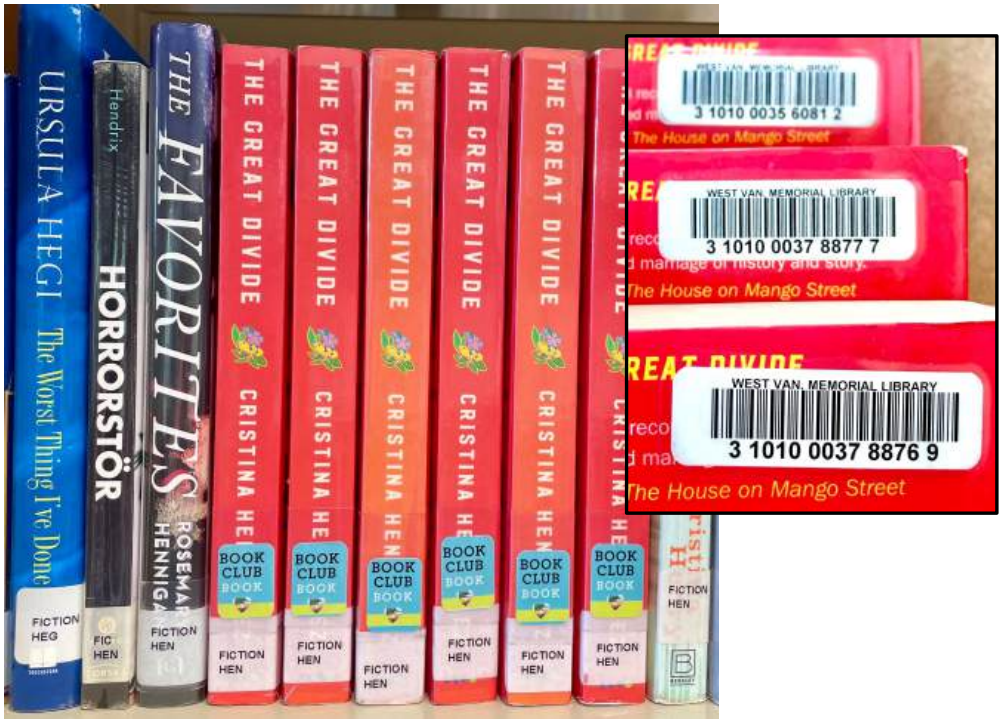
Failing to distinguish these is a common modelling error.

Alec's 2011 Volkswagen GTI

An "instance" entity – Vehicle

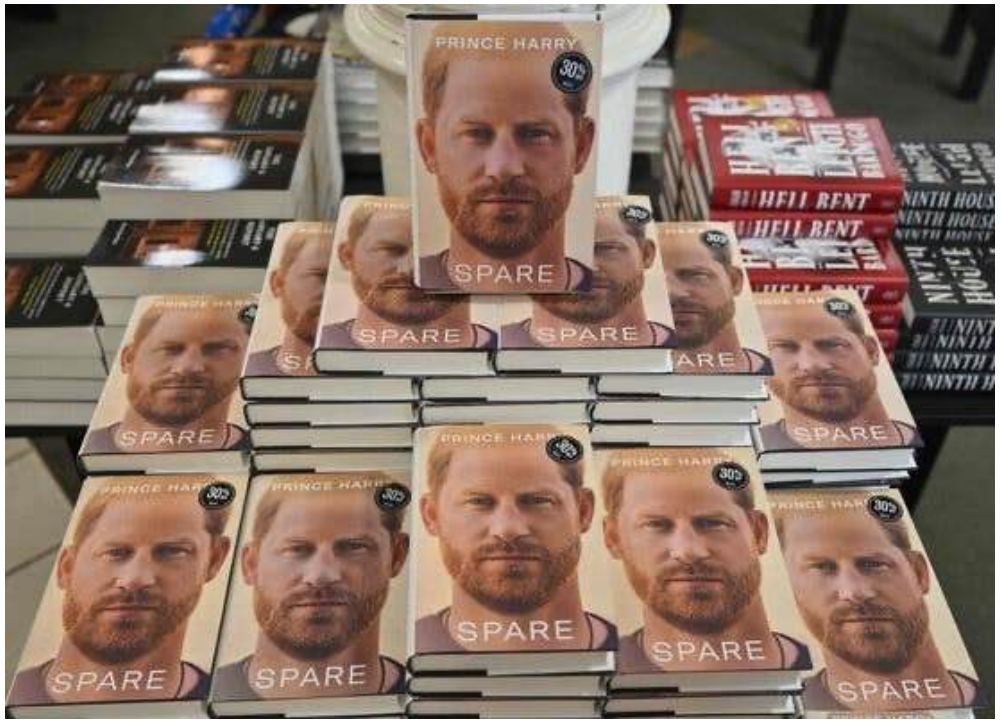
Another "Types vs Instances" example

Library – tracks *types* and *instances*



Each Copy (instance) of the same Title (type) has its own unique serial number – so we know which Cardholder has each.

Bookstore – only tracks *types*



Each instance of "Spare" does not have its own unique serial number – we can't tell one from another. All we know is how many we have. ("Quantity On Hand.")

Types vs. Instances – "What do you mean by a Bus?"

A category of Bus – a "meta-Type?"
(transit, articulated, intercity, minibus, …)
A Make and Model of Bus – a Type?
An individual Vehicle? – an Instance?

Model	Length	Width	Introduced
Xcelsior[18]	35 feet (11 m) 40 feet (12 m) 60 feet (18 m)	102 inches (2.6 m)	2008
MiDi	30 feet (9.1 m) 35 feet (11 m)	96 inches (2.4 m)	2013

"What do you mean by a Bus?"

254 British Properties

Inbound From Glenmore and Bonnymuir via Bonnymuir, Stevens, Taylor Way to Park Royal terminus (extends to Downtown Vancouver during Monday-Friday peak hours).

Outbound From Park Royal (from Downtown Vancouver during Monday-Friday peak hours) via Marine Drive, Park Royal South, Taylor Way, Southborough, Eyremount, Cross Creek, Chartwell, Crestwell, Eyremount, Fairmile, Southborough, King Georges Way, Robin Hood, Kenwood, St. Andrews, Bonnymuir to Glenmore terminus.

Park Royal to British Properties and return to Park Royal

MONDAY TO FRIDAY

Connecting Buses Leave Downtown Vancouver	Leave Park Royal	Leave Eyremount at Highland	Leave Bonnymuir at Glenmore	Leave Eyremount at Highland	Leave Marine at 14th	Arrive Park Royal	Arrive Downtown Vancouver Connecting Buses
6.35	6.53R		7.03	7.15	7.31	7.34	7.54
6.45	7.23R		7.33	7.45	8.01	8.04	8.24
7.47	8.07R		8.17	8.28	8.44*	8.47	9.16
8.20	8.40	8.53	9.06		-	9.15P*	9.41
9.22	9.47P	10.00	10.13		-	10.22P*	10.43

254 British Properties

A Bus Route?
A Bus Route Scheduled Departure
An instance of a Bus Route Scheduled Departure?

Never be afraid to ask "What do you mean by…?"

Discussion – good Entity or not?

Which of the following might ***not*** be valid entities?
And if not, *why* not?

Transcript	Student	Building	Student Directory	Faculty Member	Instructor History
Department	Course	Organisation Chart	Prerequisite List	Payment	Student Body
Class Roster	Scholarship	Faculty	Assistant Dean	Admission Date	Phillips Building
Registration	Section	Course Catalogue	Physics	Class	Professor
Admission Request Form					

And a bonus… Time

Discussion – good Entity or not?

Which of the following might ***not*** be valid entities?
And if not, *why* not?

- ✗ Transcript — a report
- ✓ Student
- ✓ Building
- ✗ Student Directory — a report
- ✓ Faculty Member
- ✗ Instructor History — a list, "history" is not singular, and a history of *what?*
- ✓ Department
- ✓ Course
- ✗ Organisation Chart — a visual report
- ✗ Prerequisite List — a list
- ✓ Payment
- ✗ Student Body — not singular
- ✗ Class Roster — a report
- ✓ Scholarship
- ✓ Faculty
- ✗ Assistant Dean — a Job Title
- ✗ Admission Date — an attribute
- ✗ Phillips Building — an instance
- ✓ Registration
- ✓ Section
- ✗ Course Catalogue — a report
- ✗ Physics — an instance
- ✓ Class
- ✗ Professor — a Job Title
- ✗ Admission Request Form — a form (artifact)
- ✗ Time — not a distinct thing ("what is a *time?")*
  but can you think of any time-related entities?

Entity definition – bad example then a good format

Customer
We have a variety of Customers that operate in multiple geographies, and these must be tracked in order to consolidate purchasing statistics and enable our rating process to identify our best Customers.

Not a good definition

- Interesting background and miscellaneous points
- *Doesn't* answer the question "What is *one* of these things?"

Customer
1. A Customer is a person or organisation that is a past, present, or potential user of our products or services.

2. Current examples include Solectron (contract manufacturer,) Cisco Systems (OEM,) Arrow Electronics (distributor,) Best Buy (retailer,) M&P PCs (assembler,) and individual consumers.

3. Excludes the company itself when we use our own products or services but includes cases where the Customer doesn't have to pay (e.g., a charity.)

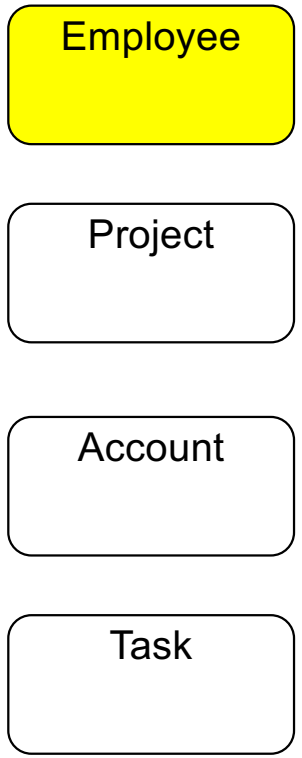
Entity definition format:

1. A description of which real-world things will be included in scope.
   This might be developed from a list of standard "thing types" – person, organisation, request, transfer, item, location, activity, etc.
   Be sure to identify any specific inclusions ("This includes…" or "This is…")
2. Illustrate with examples:
   - 5 – 10 sample instances
   - diagrams or scenarios
   - illustrations such as reports or forms
3. Interesting points – anomalies, synonyms, common points of confusion, etc.
   May include specific exclusions ("This excludes…" or "This is not…")

Discussion – starting an Entity definition

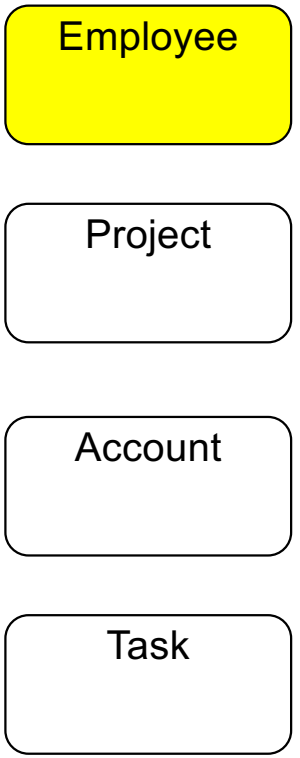
*"Can anyone think of examples that might surprise someone else –
that is, anomalies or potential sources of confusion."
E.g., how could we legitimately have different ideas what "Employee" means?*

-
-
-
-
-
-
- ...



Brainstorming space

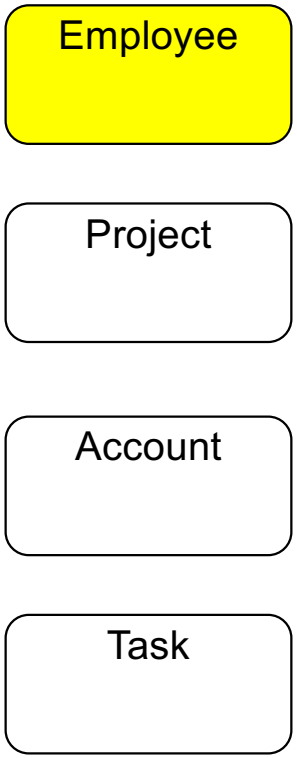
*"Can anyone think of examples that might surprise someone else –
that is, anomalies or potential sources of confusion."
E.g., how could we legitimately have different ideas what "Employee" means?*



Starting an Entity definition

"Can anyone think of examples that might surprise someone else – that is, anomalies or potential sources of confusion."
E.g., how could we legitimately have different ideas what "Employee" means?

F/T vs. P/T?	– *Both*
Only IS Department?	– *No*
Include management, or only individual contributors?	– *Yes, everyone*
Still in recruitment (an applicant)?	– *No*
Onboarded? on probation? active? retirees?	– *Yes, all*
Include contractors, student interns, vendor staff, etc.?	– *Yes, all*
Volunteers?	– *Yes*
A type of worker (DBA or tester) or a specific person?	– *No, only a specific person*
A robotic, automated, or AI agent?	– *No, only a real person*



Defining the Entity "~~Employee~~" – "Worker"

Definition format:

1. A description of which real-world things are within in scope, and any specific inclusions ("This *includes*…" or "This *is*...")

2. Illustrate with examples – 5 to 10 sample instances or types

3. Interesting points – anomalies, synonyms, common points of confusion, etc. May include specific exclusions ("This *excludes*…" or "This *is not*...")

Worker (renamed from Employee):

A *Worker* is a person, whether or not directly employed by *the company,* but with some sort of employment contract or arrangement, who has been or may be assigned to a Project.

Worker includes:

- Full or Part-time Employees who have been onboarded, including Probation, Active, Seconded, Suspended, Retired…
- Contractors
- Consultants
- Student Interns
- Vendor Staff Persons
- Company Owners and Managers

Key points:

- "Worker" was chosen as the entity name because it is more generalised than "Employee."
- A Worker may not necessarily be billable on a Project, e.g., a non-chargeable Subject Matter Expert or Volunteer
- Worker excludes:
  - Job Roles, e.g., DBA or Technical Writer
  - Robotic, Automated, or AI Agents (this might change)

Another example – starting an entity definition for Task

"Can anyone think of examples that might surprise someone else – that is, anomalies or potential sources of confusion."
E.g., how could we legitimately have different ideas what "Task" means?

-
-
-
-
-

Worker

Project

Account

Task

Another example – starting an entity definition for Task

"Can anyone think of examples that might surprise someone else – that is, anomalies or potential sources of confusion."
E.g., how could we legitimately have different ideas what "Task" means?

Key points that typically arise:

- A *type* of Task or a *specific* Task? (the types vs. instances problem)
- Part of a *specific* Project or used across *multiple* Projects?
- Produces a specific *deliverable* or *state*?
- Time-bounded or ongoing?
- Performed by *one* Worker or *one or more* Workers?
- …

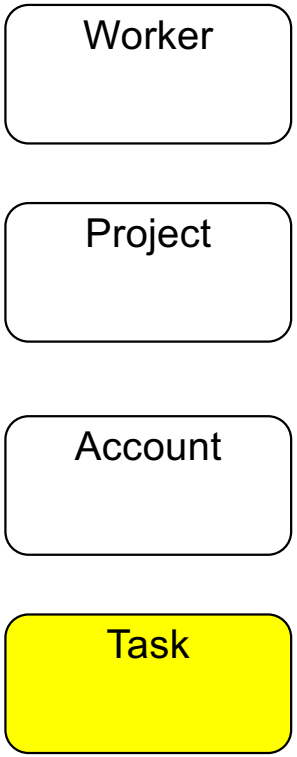
A ***Task*** is a specific, time-bounded, unit of work, within a single Project, intended to be performed by one or more Workers, that produces an intended deliverable or achieves a specific state.

Examples:

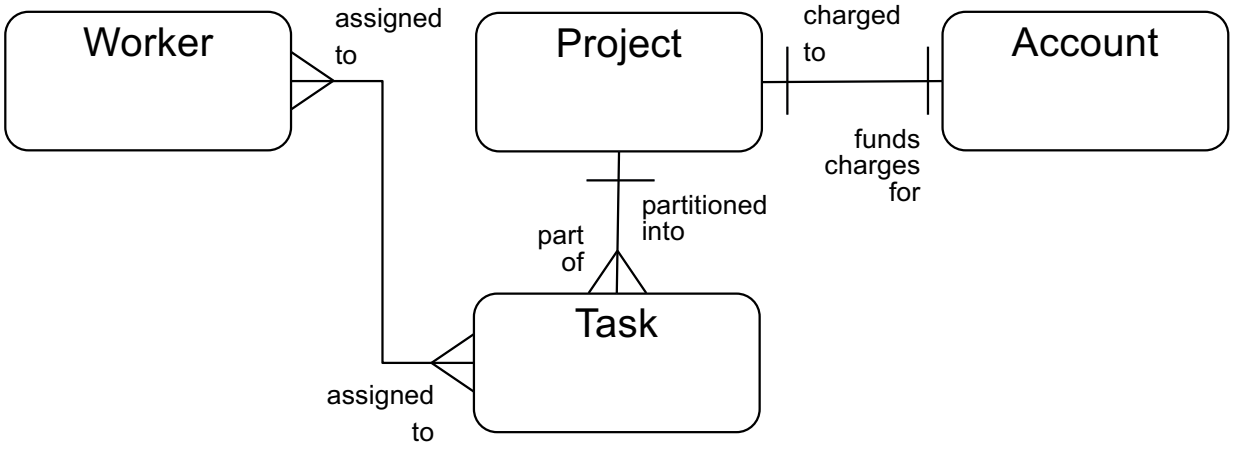
- Code *Place Order* service
- Test *Place Order* service

Excludes:

- types of Tasks
- ongoing (non time-bounded) activities such as management or administration



Now we have definitions – it's "safe" to draw the ER model



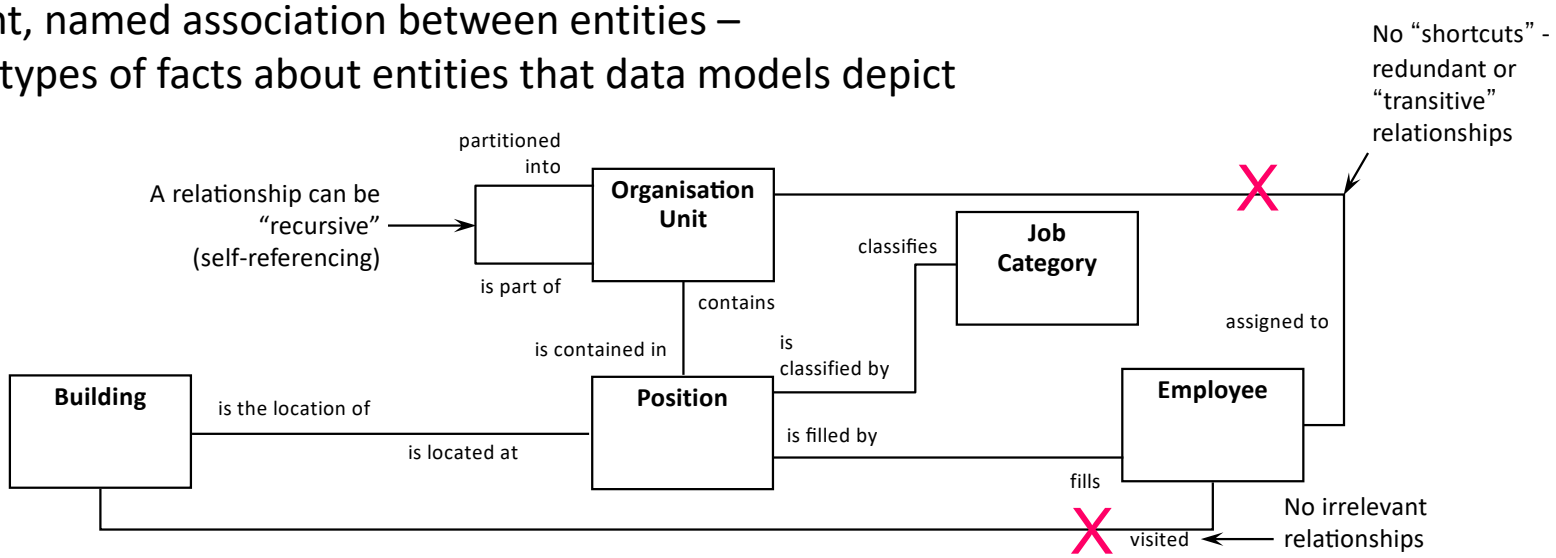
First arrange entities top-down by dependency.

Then add relationships with a verb-based phrase.

Then add cardinality (1:1, 1:M, M:M.)

Relationships – a few more points

A significant, named association between entities –
one of the types of facts about entities that data models depict



Guidelines

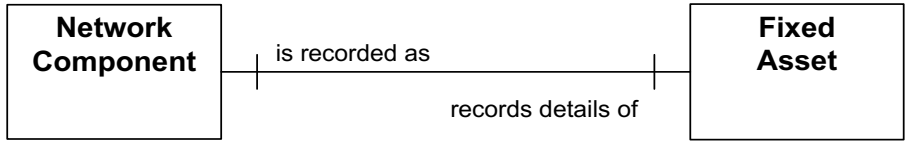
- named with a descriptive, verb-based phrase – not "has" or "is related to"
  (the line tells us they *are* related; the name tells us *how*)

- named in both directions – try to use the same root word at both ends
  (e.g., "classifies" and "is classified by")

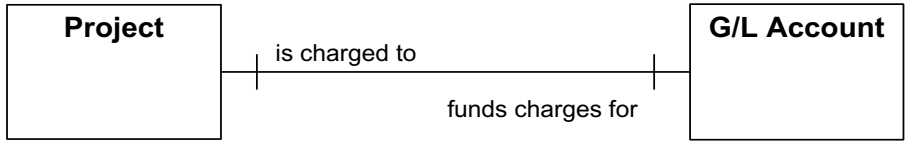
- the complete name reads like a sentence (noun verb noun) –
  "Position is classified by Job Category"

1:1 relationships – almost always an error!

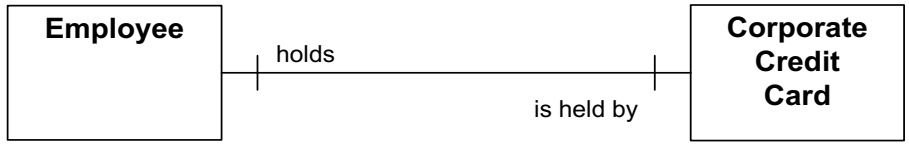
- Note – a 1:1 relationship might be necessary in the Physical Database Design
  e.g., "Fixed Asset" records financial data about a "Network Component" but they are in two separate systems (the G/L System and the Configuration Management System) connected by a 1:1 relationship



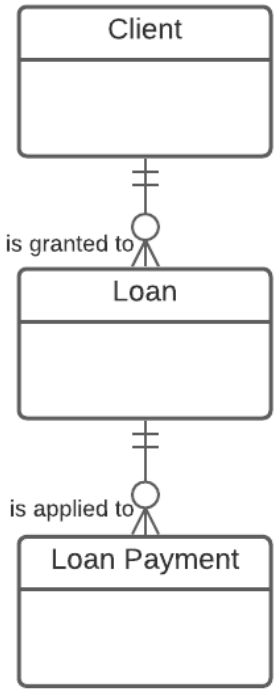
- ✘ Incorrect analysis
  e.g., Project costs are probably prorated across *many* Accounts



- ✘ Failing to account for changes over time
  e.g., an Employee may hold only *one* Credit Card at a time, but *many over time,* and we virtually always want history.
  The most common written constraint in Concept Modelling is
  "one <u>at</u> a time but many <u>over</u> time."

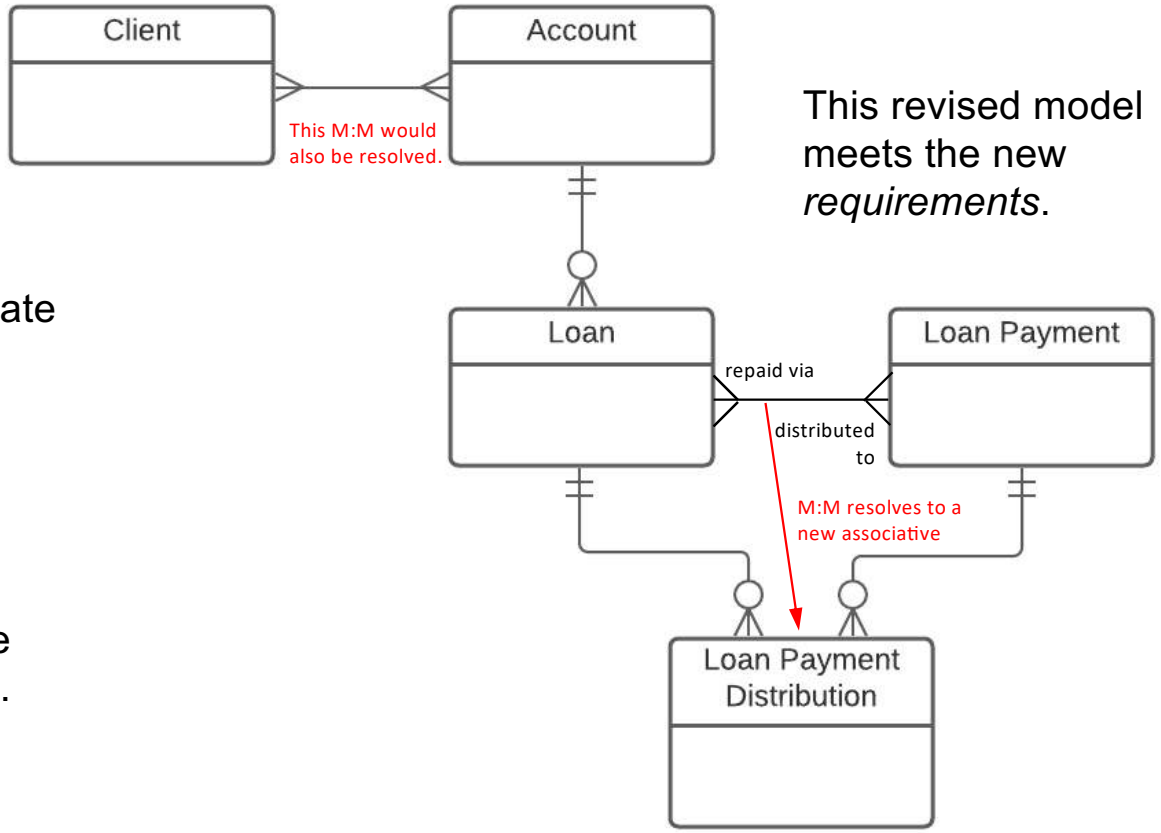


Future-proofing – "Challenge the Ones"



A Loan is granted to
one and only one Customer –
really?
No – multiple Clients can participate
in a Loan via a shared Account.
(A new *requirement*.)

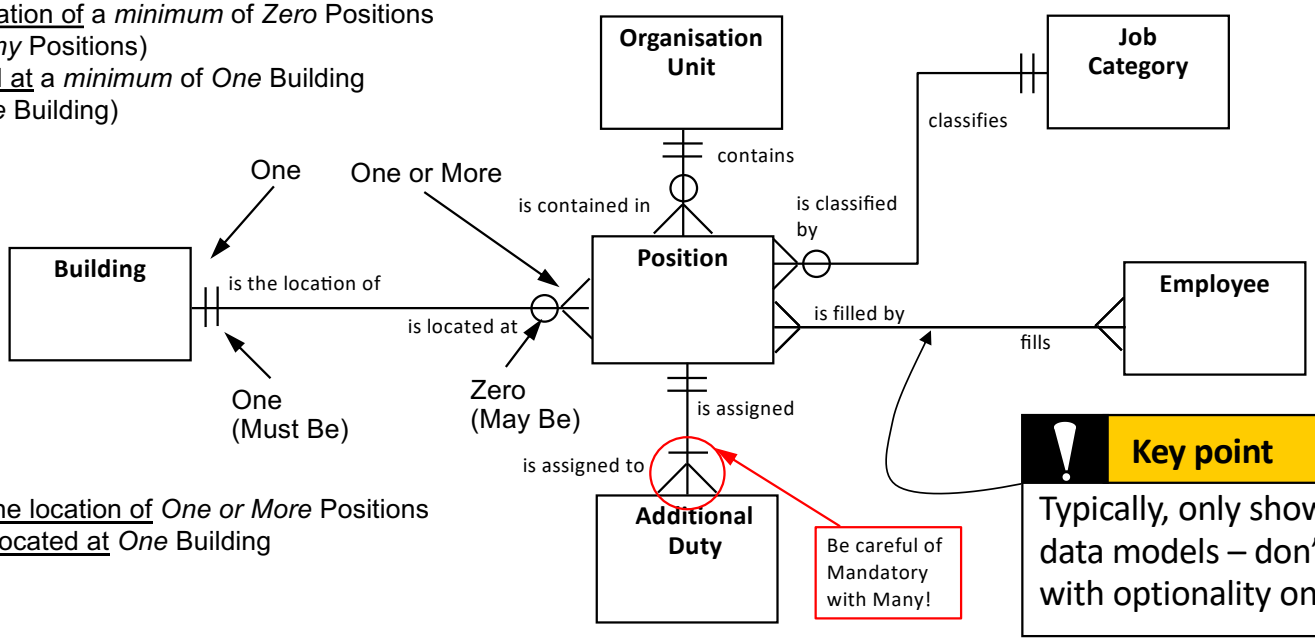
A Loan Payment applies to
one and only one Loan –
really?
No – one Loan Payment could be
distributed across multiple Loans.
(A new *requirement*.)



This revised model
meets the new
requirements.

Relationship optionality (logical models only)

Each Building is the location of a *minimum* of *Zero* Positions
(and a *maximum* of *Many* Positions)
Each Position is located at a *minimum* of *One* Building
(and a *maximum* of *One* Building)



or,
Each Building *May Be* the location of *One or More* Positions
Each Position *Must Be* located at *One* Building

To determine optionality (a.k.a. minimum cardinality)

- for each entity ask "Can one of these be related to a *minimum* of *Zero* or a *minimum* of *One* of the other entity?"
- record the answer – 0 or 1 – at the "other" end
  "zero" means an optional relationship (*May Be*) and
  "one" means a mandatory relationship (*Must Be*)
- easier form: "Each one of these *May Be* be or *Must Be* related to the other?"

Don't forget the four Ds of Data Modelling

1 Definition

- "What *is* one of these things?"
- List common and unusual instances
- "Are there any known anomalies?"
- "What are the potential differences of opinion?"

2 Dependency

- "What type of entity is this?"
- "What other entity does it depend on?"
- Essentially
  - is it a free-standing thing?,
  - is it a type of thing?,
  - is it repeating detail about some other thing?

Please let us know the key point (or points) that mattered most to you in this first section.

3 Detail

- Don't dive into detail – keep it in its place!
- GEFN!* HPDL!**

*Good enough for now!
**Hard part, do later!

4 Demonstration

- Assertions / narrative rules
- Sample data values or instances
- Scenarios or use cases
- Props (e.g., report layouts or common documents)

Phase 2 of three phases in data modelling

1) Establish initial Concept Model

- Focus is on developing a core set of entities:
  - named
  - defined
  - minimally attributed
  - bound by basic rules and relationships
  - placed on an ERD
- Might start bottom-up: brainstorm details then synthesising "up"
- Might start top-down: build a contextual model, then flesh out required details analysing "down"
- Experiment w. alternatives
- Refine the contextual model, if you had one.

2) Develop initial Logical Data Model

- Focus shifts to attribute rigor and structure when going to the logical level
- First check attributes for:
  - completeness
  - necessity
  - name and definition
  - placement
- Resolve attributes that are:
  - multi-valued
  - redundant
  - constrained
- Continue experimenting with alternate structures
- Refine conceptual model

3) Refine & extend Logical Data Model

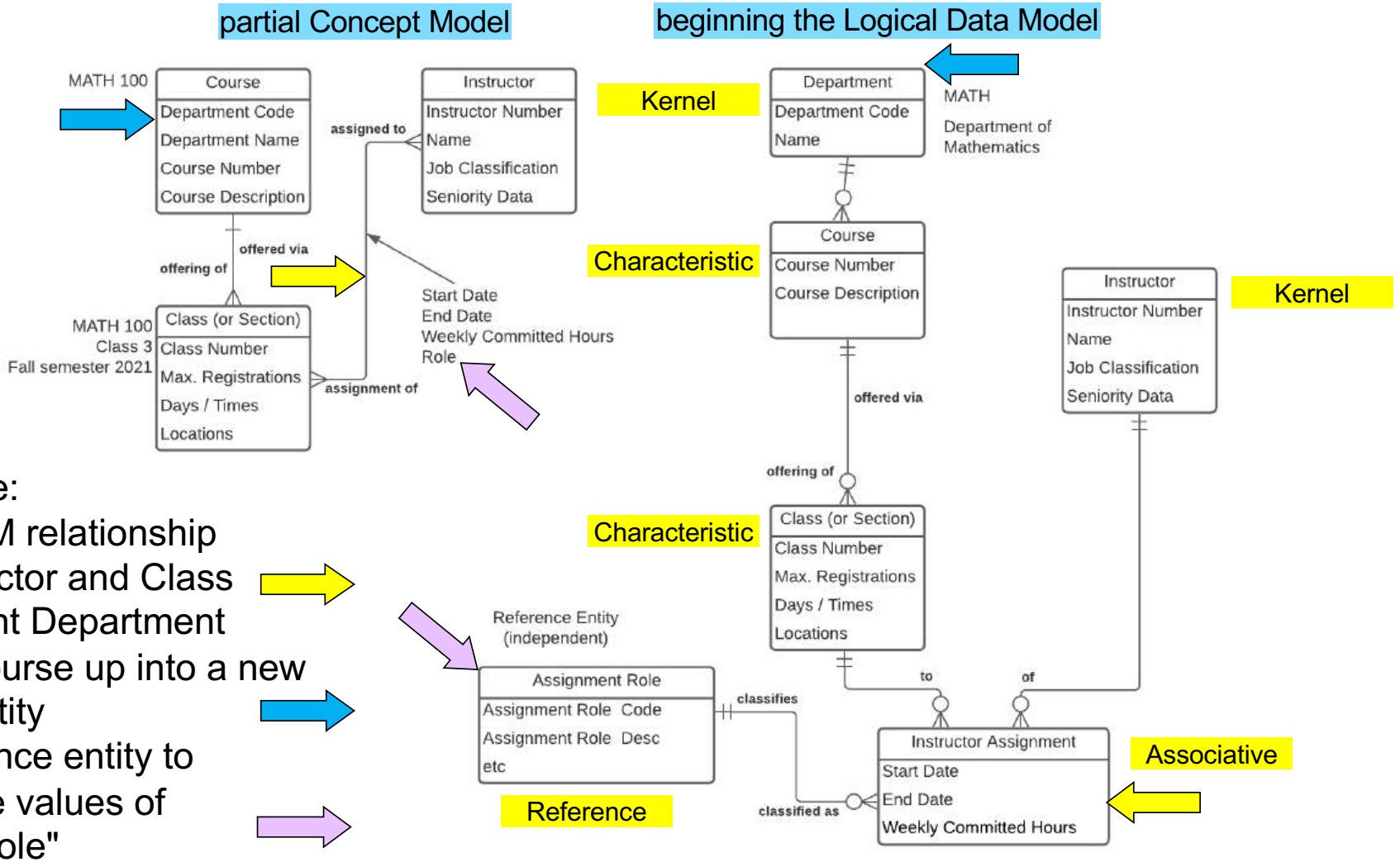
- Focus is on refinement, and validation via new requirements using…
- …an event-based approach: fast and easy…
- …or full business analysis:
  - process workflow model
  - use cases (external)
  - service specs (internal)
  - profiling existing data
  - informational needs
- Resolve attributes that are semantically overloaded, non-atomic, or derived
- Document attribute properties and validation
- Specify identifiers
- Refine conceptual model

From conceptual to initial logical

The progression from conceptual to logical is largely based on identifying and dealing with three attribute characteristics

- Multi-valued - the attribute can have multiple different values for one instance of the entity, either "at a time" or "over time"
  E.g., "Employee Name" if aliases or previous names are tracked
  - move it **down** to the "many" end of a 1:M relationship into a characteristic entity
  - if it's a fact about a M:M relationship between entities, move it down to the "many" end of a 1:M relationship into an associative entity
  - this puts the data structure into 1st Normal Form – 1NF
- Redundant - the same attribute value is recorded multiple times, in different entity instances, possibly inconsistently
  E.g., "Company Name" in a "Department" entity
  - move it **up** to the "one" end of a M:1 relationship to one of the parent (or higher) entities (2nd Normal Form – 2NF)
  - You might have to create a new parent entity where none existed before
- Constrained - a descriptive attribute needs to be restricted to a set of standard (or "allowable") values to improve integrity and reporting
  E.g., "Employee Type"
  - move it **out** to the "one" end of a M:1 relationship to a reference or other related entity (3rd Normal Form - 3NF)

A simple Concept Model to Logical Data Model example

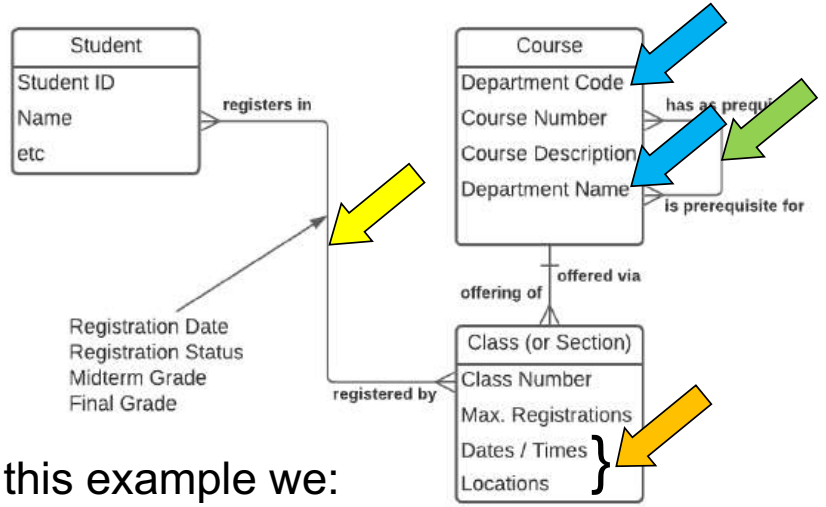


In this example we:

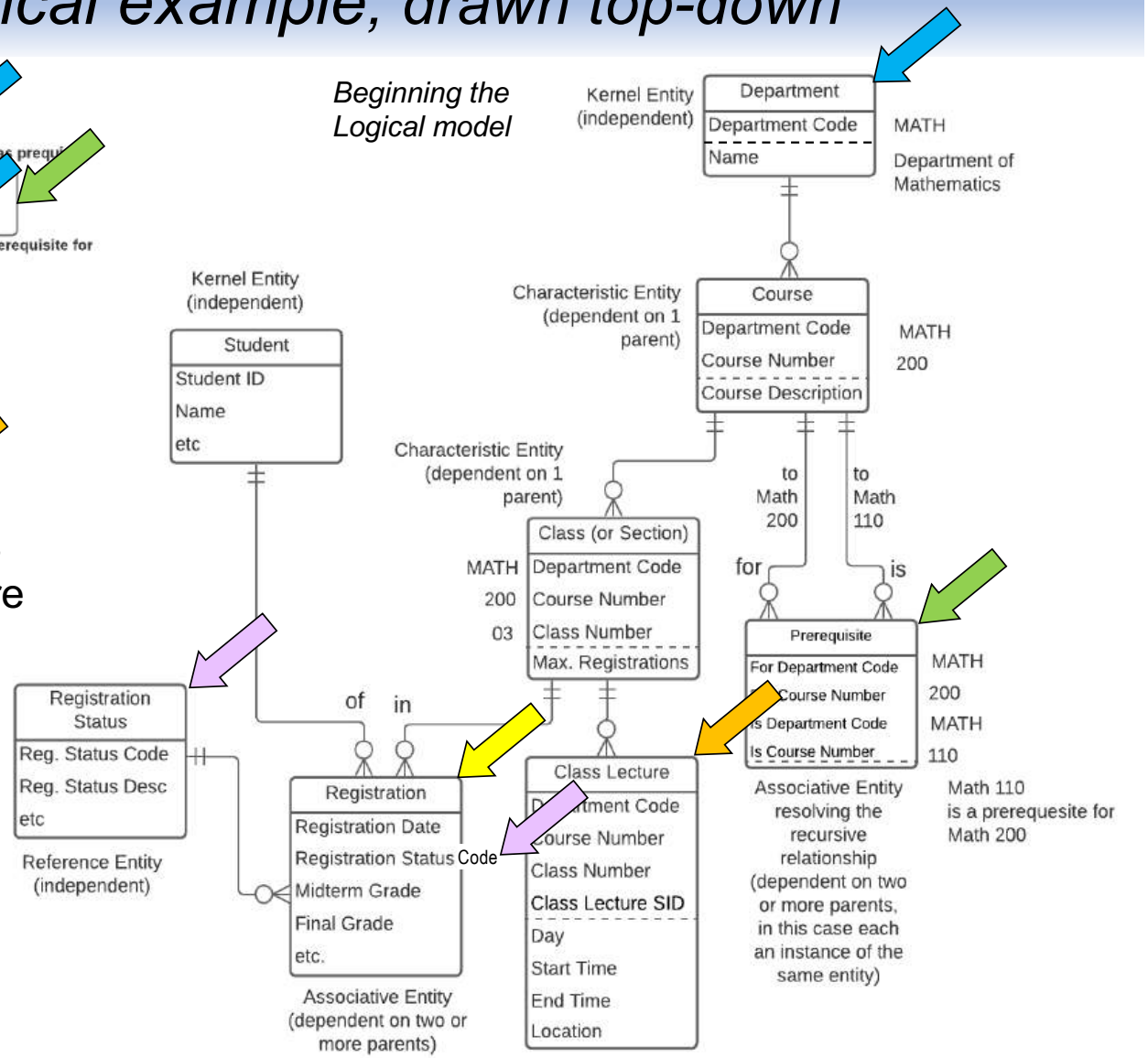
- resolve the M:M relationship between Instructor and Class
- move redundant Department attributes in Course up into a new Department entity
- create a reference entity to standardise the values of "Assignment Role"

Another Concept to Logical example, drawn top-down

Conceptual



Beginning the Logical model



In this example we:

- move multi-valued Class attributes into their own entity – Class Lecture
- resolve the M:M relationship between Student and Class
- resolve the recursive Course to Course M:M relationship
- move redundant Department attributes in Course up into a new Department entity
- move Registration Status into a reference entity

World's shortest course on normalisation

Unnormalised (UNF or 0NF)

- Contains multivalued attributes (a "repeating group")

First Normal Form (1NF)

- Repeating attributes moved *down* to a dependent Characteristic or Associative entity (create a new dependent entity if necessary.) This makes data "reportable."

Second Normal Form (2NF)

- Only applies to dependent entities
- No attribute in a child entity is really a fact about a parent (or grandparent or…)
- That is, no Characteristic or Associative entity redundantly contains facts from its parent(s) – if it does, move the fact(s) *up* (create a new parent entity if necessary)

Third Normal Form (3NF)

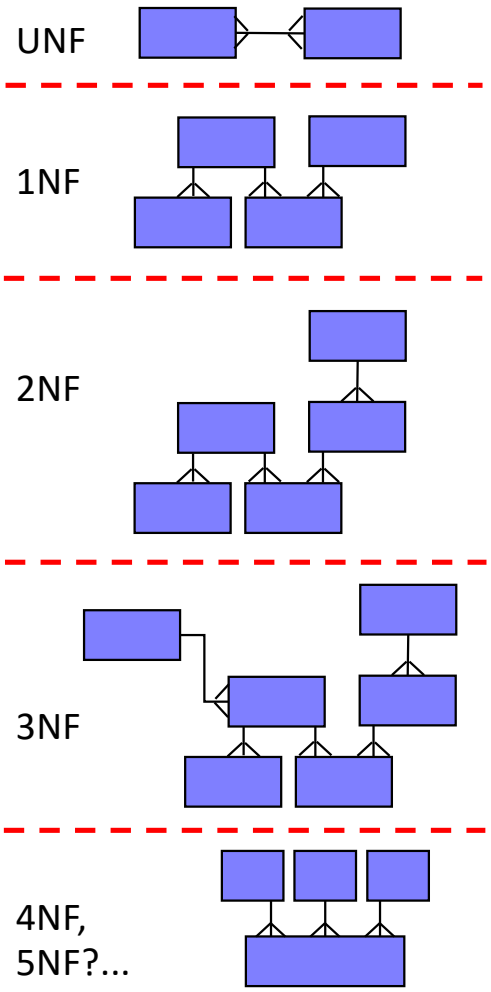
- If any entity redundantly contains facts from a related (non-parent) entity, move the fact(s) *out* to the other entity (create a new entity if necessary)

BCNF (Boyce-Codd NF – "3.5NF")

- Not an issue if you keep your wits about you

Fourth and Fifth Normal Form (4NF, 5NF)

- "Large" (3-way or more) associatives need to be broken down into more granular entities



For reference – Contextual, Conceptual, & Logical models

1 Contextual (Scope – Planner's View)

Agree context or "big picture" – the scope in terms of topics or subjects that are in or out, plus core terms and definitions

- May be a simple block diagram of topics/subjects, or primarily textual (a list)
- Optional – not necessary on smaller projects

My most plagiarised slide ever!

2 Conceptual (Overview – Owner's View)

Agreement on basic concepts and rules

- Ensures everyone is using the same vocabulary and concepts before diving into detail
- Overview: main entities, attributes, relationships, rules
- Lots of M:M relationships
- Relationships show cardinality
- No keys
- Few or no reference entities
- Unnormalised – most M:M relationships unresolved, many attributes will be multi-valued, redundant, and non-atomic
- Verified directly by clients plus other techniques: Use Cases…
- A "one-pager"
- 20% of the modelling effort

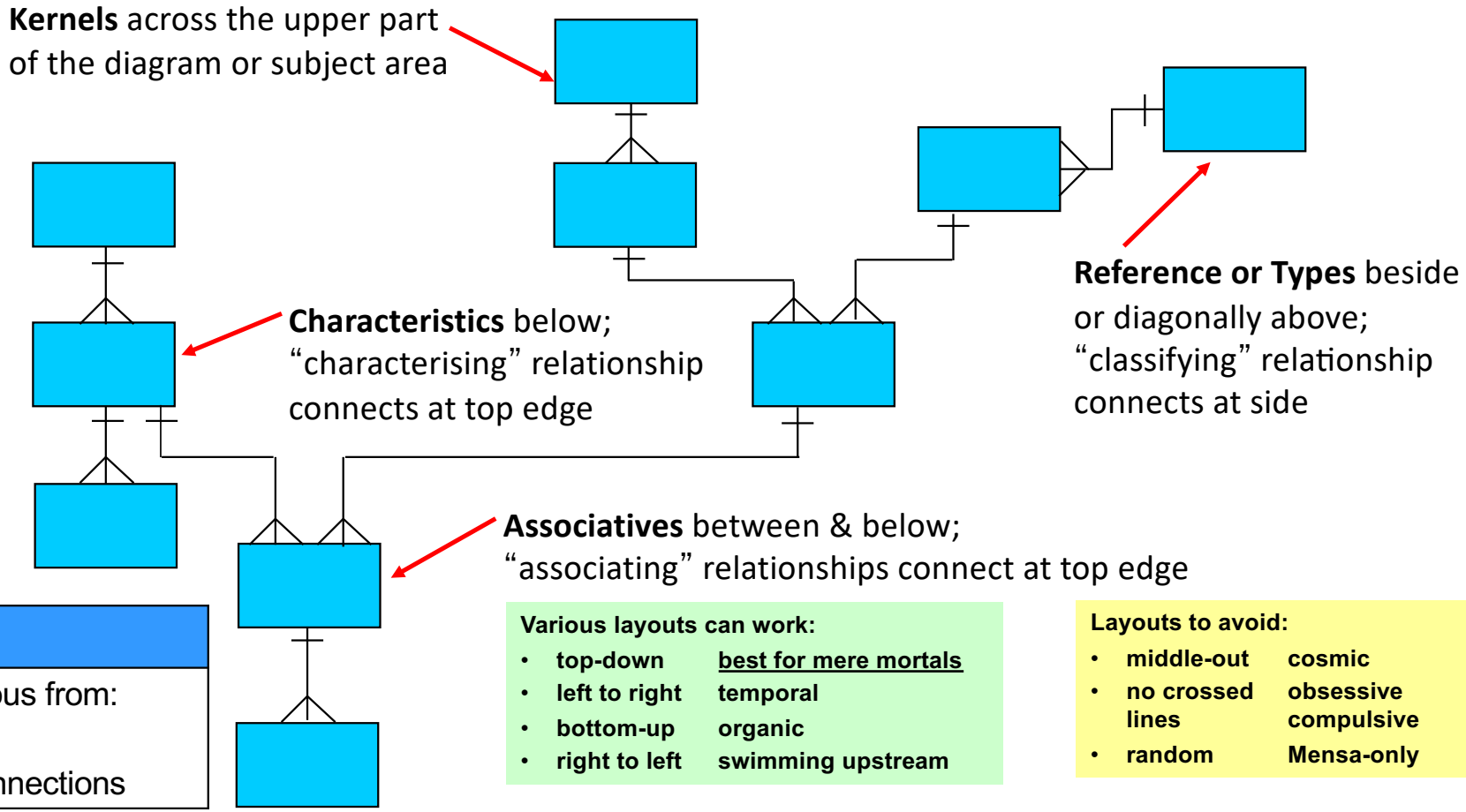
3 Logical (Detail – Designer's View)

Full detail for physical design

- Provides all detail for initial physical database design and requirements specification
- Detailed: ~ 5 times as many entities as the conceptual model
- M:M relationships resolved
- Relationship optionality added
- Primary, foreign, alternate keys
- Lots of reference entities
- Fully normalised – no multi-valued, redundant, or non-atomic attributes. All attributes defined and "propertised"
- Verified by other means: sample data, report mockups, scenarios, …
- May be partitioned
- 80% of the modelling effort

Graphic guidelines – the "no dead crows" principle

Draw the same kinds of things the same way every time!



! Key point

Entity type is obvious from:
- Placement
- Relationship connections

Various layouts can work:

• top-down	best for mere mortals
• left to right	temporal
• bottom-up	organic
• right to left	swimming upstream

Layouts to avoid:

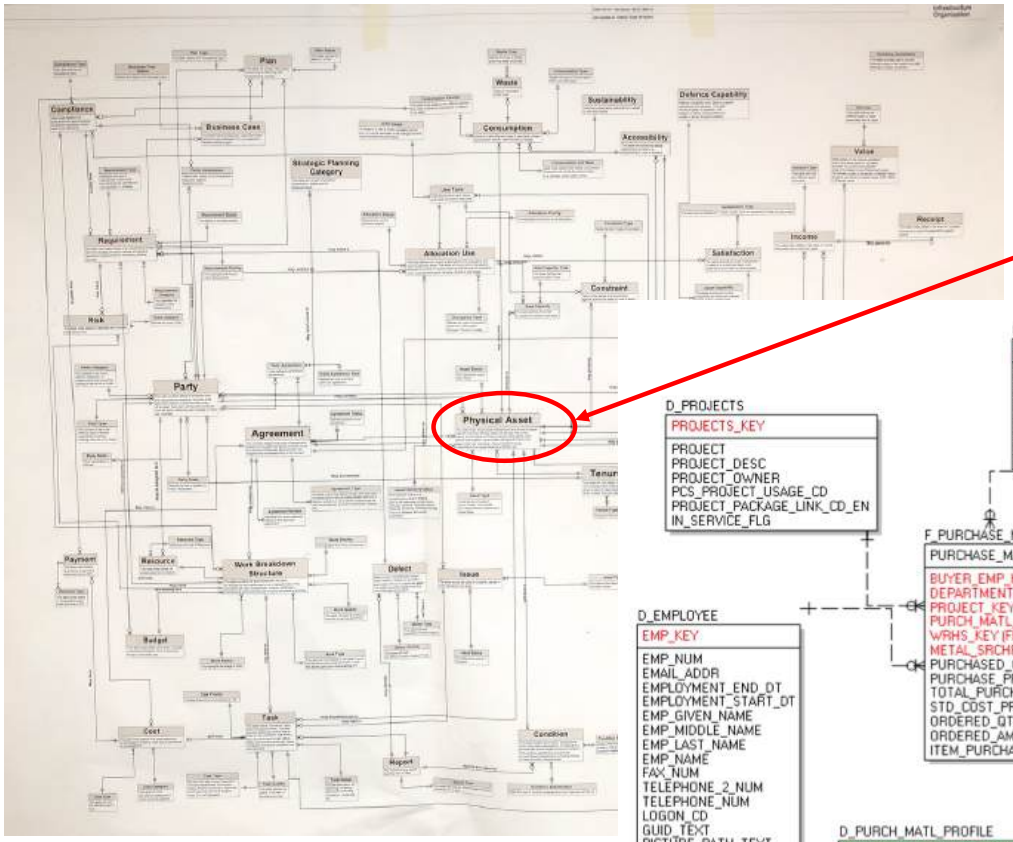
• middle-out	cosmic
• no crossed lines	obsessive compulsive
• random	Mensa-only

Note! What works for Dimensional Models doesn't for E-R Models

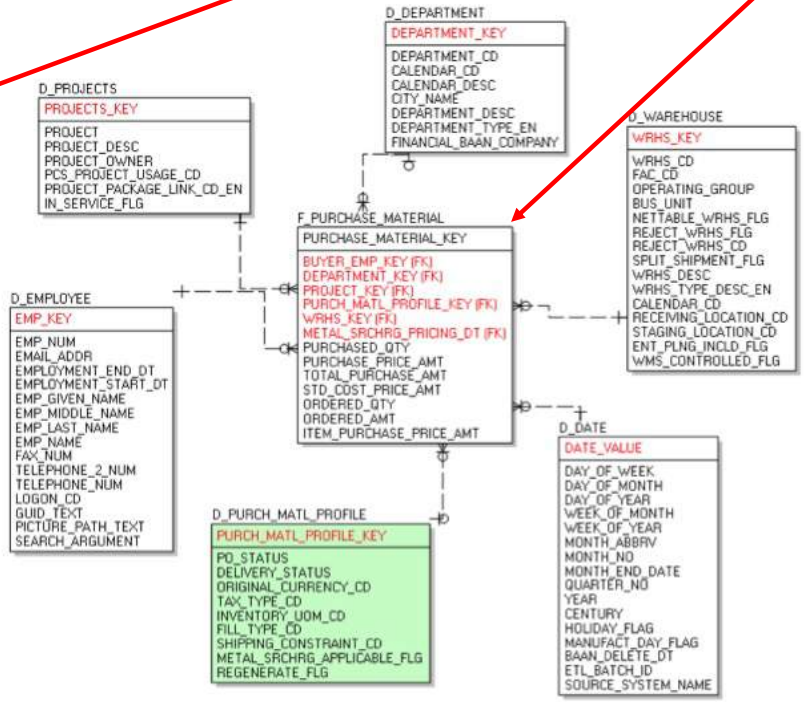
A common error –
"the most important entity should go in the centre of the diagram."

An excellent model *structurally*, but very difficult to follow –
no sense of direction.

Concept Models / E-R Models should be drawn top-down by dependency.



"Fact" in the middle - fine for Dimensional, terrible for E-R

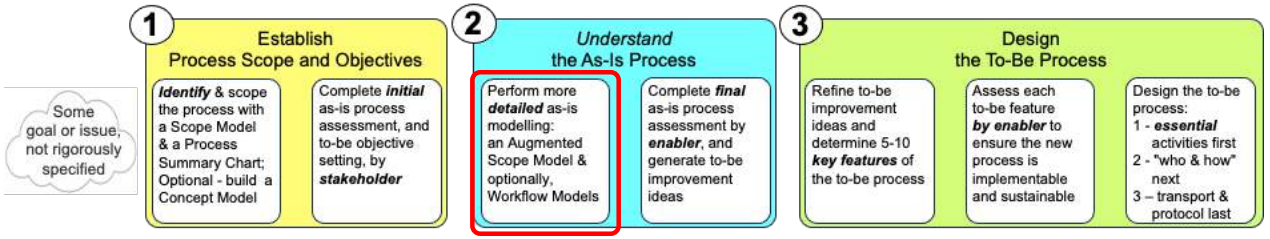


Business Process Modelling and Design

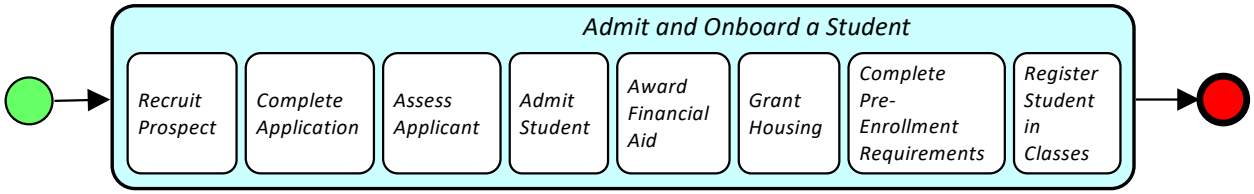
Course Topics

1. Requirements Definition
   - Goals, Issues, and the Return of Modelling
   - Case Study - Integrating the Techniques
2. Business Process Fundamentals
   - Five Things You Need to Know
   - Discovering, Scoping, & Assessing Your Processes
3. Concept Modelling Fundamentals
   - E, R, A - A Concept Model's Essential Components
   - Drawing Your Model for Maximum Understanding
4. Business Process Workflow Modelling & Design
   - Five Core Guidelines for Great Swimlane Diagrams
   - Facilitating a Process Mapping Session
   - Assessment of the As-Is and Transition to the To-Be
5. The Process-Data Connection
   - The Natural Synergy between Process & Data Models
   - Process-Data Synergies in Modelling, Analysis, & SW

Complete additional as-is modelling



The goal is to *understand* the as-is process, not document it in *excruciating detail!*



- Identify Suspects
- Qualify Prospects
- Engage Prospect
- etc.

- Collect App Fee
- Initiate Application
- Submit Application
- etc.

- Confirm Application
- Evaluate Application
- Verify Req'ts
- etc.

- Make admit / deny / assess decision
- Notify Student
- etc.

- Receive FAFSA
- Assess Need
- Determine Aid
- etc.

- Provide Housing Req'ts
- Assess Application
- Provide Alternatives
- etc.

- Confirm Other Requirements (visa, shots, writing, …)
- Register Orientation
- etc.

- Identify Courses
- Create Class Schedule
- Register Classes
- etc.

Who: Registration Assistant
What: Register Classes
How: via Workday SRS

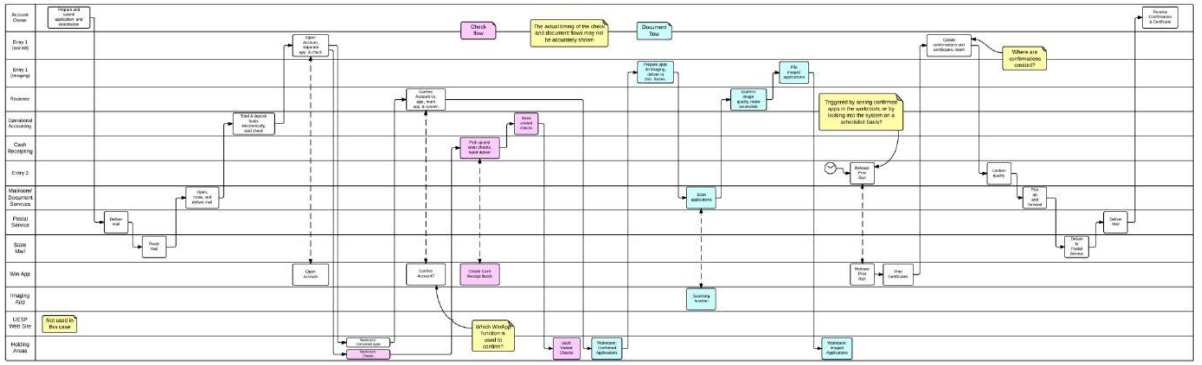
Optionally, model initial Workflow –

- Simplicity – minimal symbols and detail
- "Flow first, detail later!"

I always build an Augmented Scope Model –

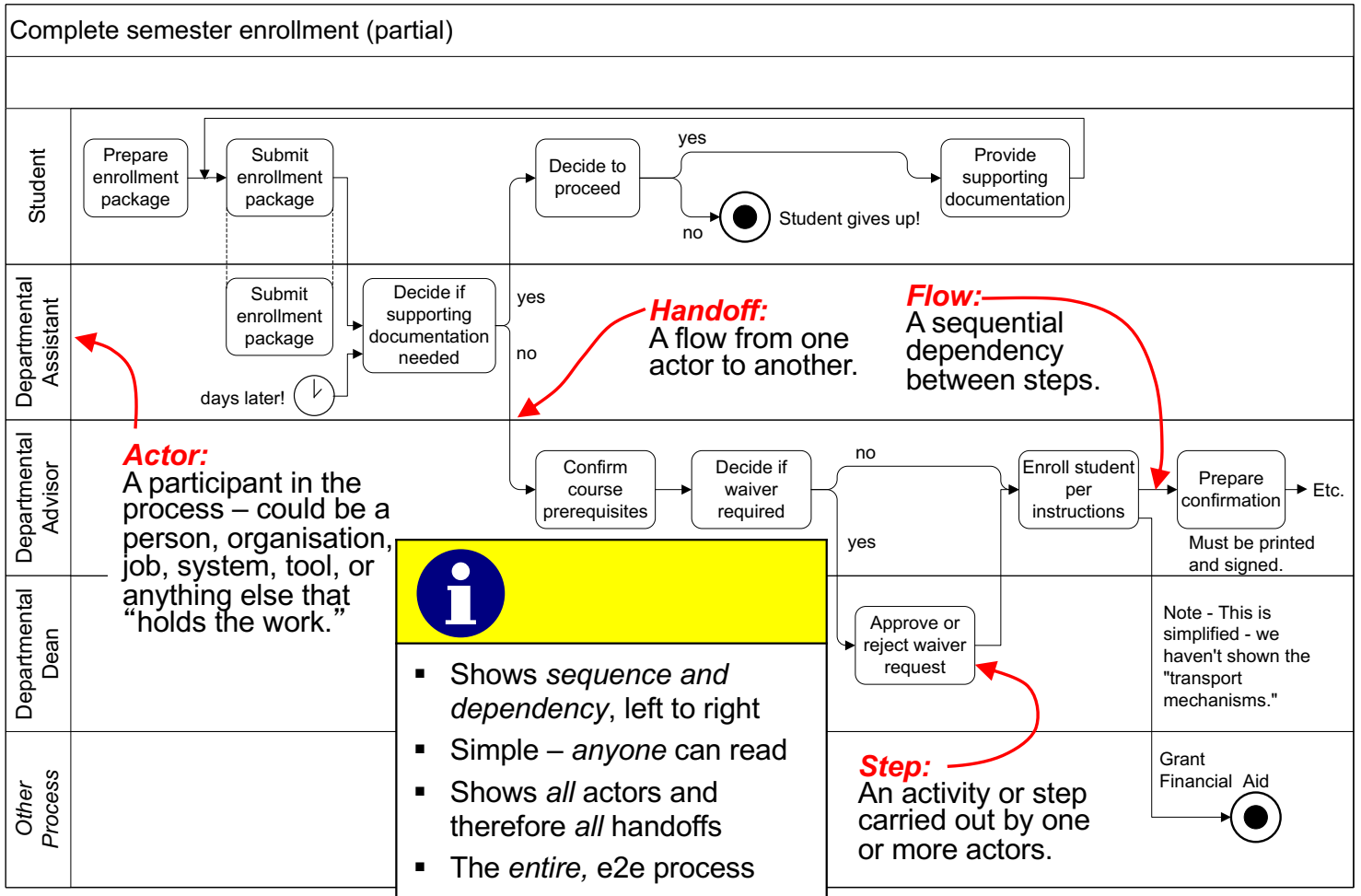
1. *What* the *detailed* activities are, e.g. "Register Class" (verb + noun)
2. Add *who* and *how*, e,g, "*Advisor* Register Class *via SIS*"

This is often good enough! – no need for an as-is swimlane diagram / workflow model



www.lucidchart.com

Simple Swimlane Diagrams – maximise their strengths



Who – the actors

What – the steps

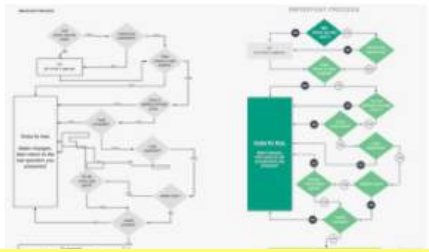
When – the flow

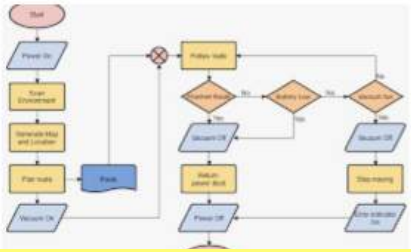
Other tools are better for capturing detail – *how* the steps are done:

- step-by-step procedures
- checklists
- decision trees
- use cases
- etc.

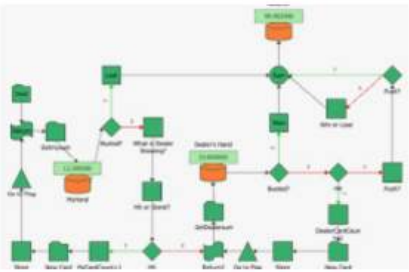
Why did simple Swimlane Diagrams become popular?

A quick Google Images search on "swimlane diagram" reveals…





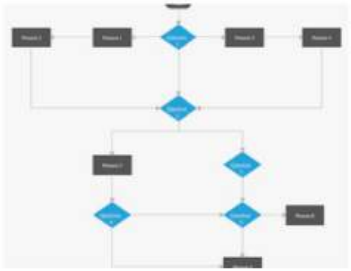
Dota 2 Flow Chart ...
reddit.com



complex RENO flowcharts easier ...
weibull.com

… lots of diagrams I might draw differently.

Follow flowchart best practices without ...
cacoo.com

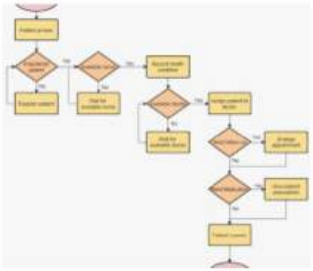


Flowchart Tutorial (Complete Flowchar...
creately.com

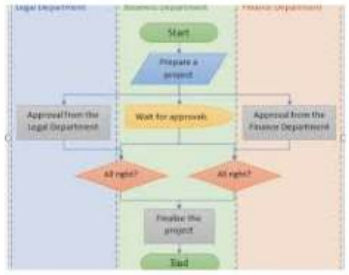
21 Creative Flowchart ...
visme.co

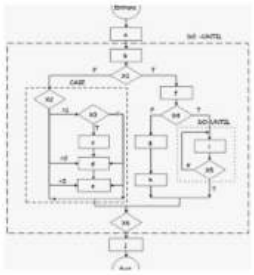
Flowchart Programming ...
conceptdraw.com

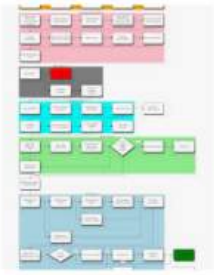
Free Flowchart Templates ...
gliffy.com

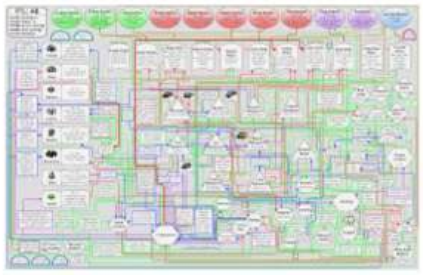


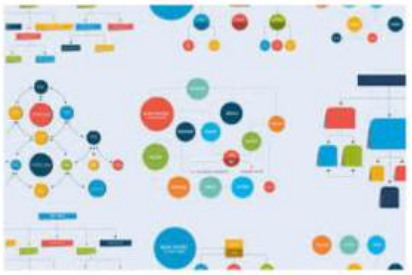
Flowchart Tutorial (with Symbols, ...
visual-paradigm.com

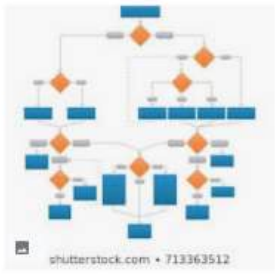




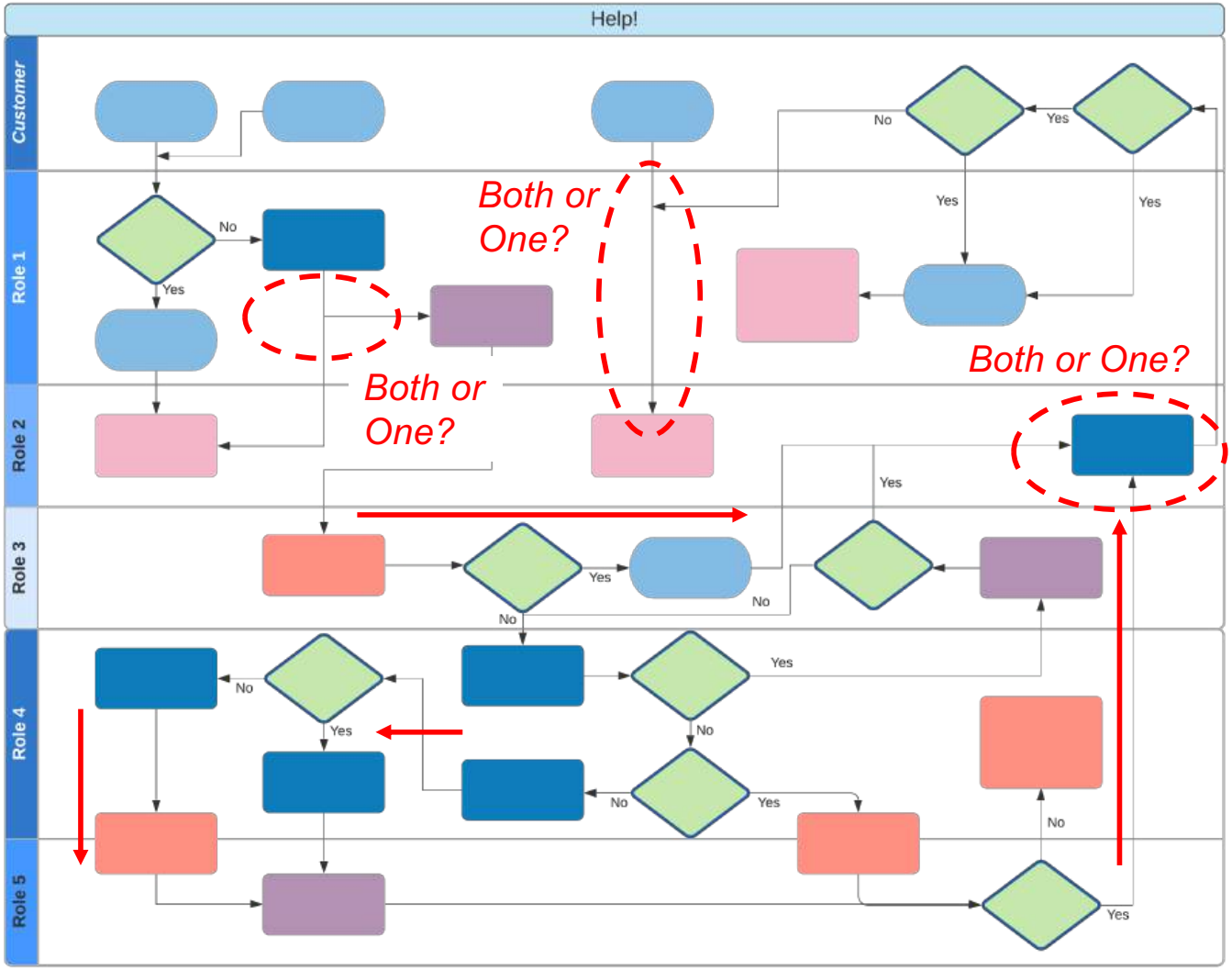








One example… "Chaos With Colours"



Probably accurate,
not too many symbols, but…

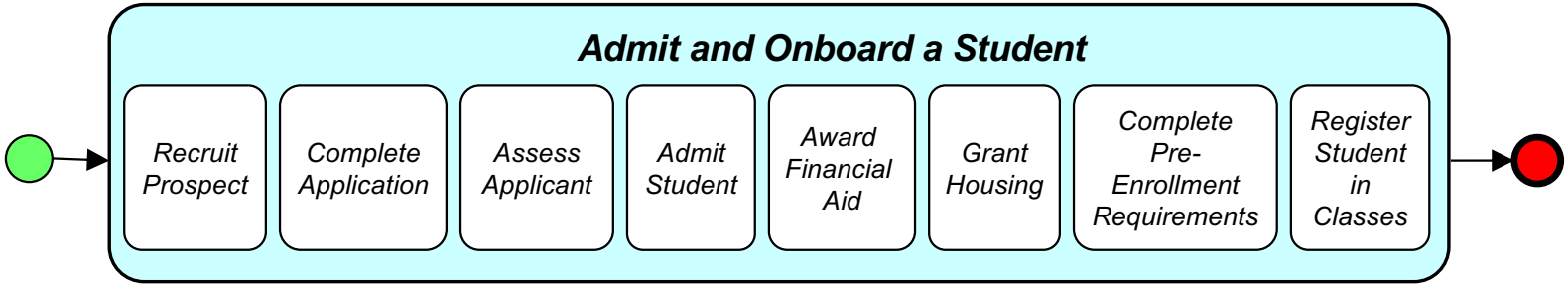
- do unexplained colours help?
- significance of multiple flows?
  - two separate flows inbound to a step
  - two joined flows inbound to a step
  - one outbound flow splitting
- but most of all…
  flows in all directions!:
  - left to right
  - right to left
  - top down
  - bottom up

Why???

Forcing it into a "one-pager" defeats the graphic power of the diagram.

If you need a one-pager draw an Augmented Scope Model

Add 5 – 10 Activities per Major Activity



- Identify Suspects
- Qualify Prospects
- Engage Prospect
- etc.

- Collect App Fee
- Initiate Application
- Submit Application
- etc.

- Confirm Application
- Evaluate Application
- Verify Req'ts
- etc.

- Make admit / deny / assess decision
- Notify Student
- Complete Integrated Assessment
- etc.

- Receive FAFSA
- Assess Need
- Determine Aid
- etc.

- Provide Housing Req'ts
- Assess Application
- Provide Alternatives
- etc.

- Confirm Other Requirements (visa, shots, insurance, …)
- Register Orientation
- Complete Orientation
- etc.

- Identify Courses
- Create Class Schedule
- Register Classes
- etc.

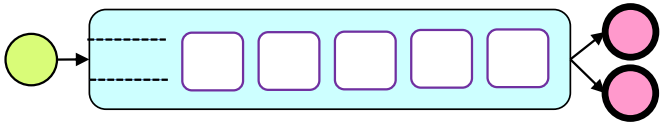
Later add *who* & *how* to each Activity

Who: Registration Assistant
What: Register Classes
How: via Workday SRS

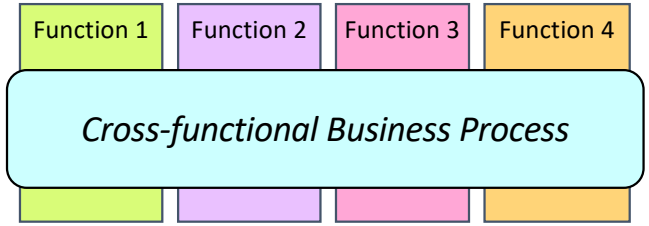
Before "swimlaning"…

We're almost at swimlane level!

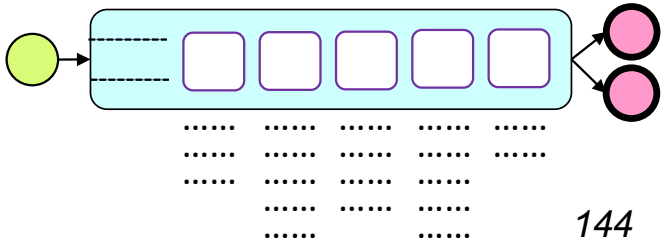
Always!
Process Scope Model *(TRAC)*



Always!
Process Summary Chart



Almost always!
Augmented Scope Model

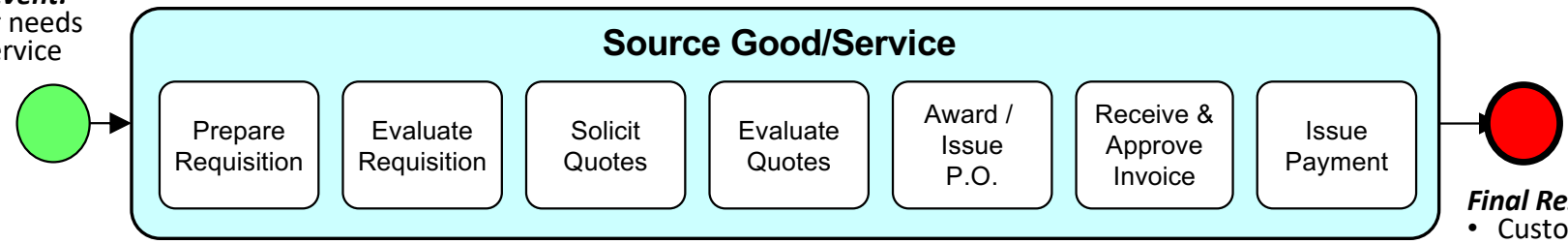


Another fast Augmented Scope Model example

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

Assume everything <$5000 is purchased with a PCard

Triggering Event:
- Customer needs Good / Service



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)

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

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.

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

Generate Purchase Order

Notify Requestor

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

Receive Good/Service
* Invoice could be attached

Accept Good/Service

Issue invoice (vendor)

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

Issue Payment (Magic Happens Here)

Receive payment

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

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

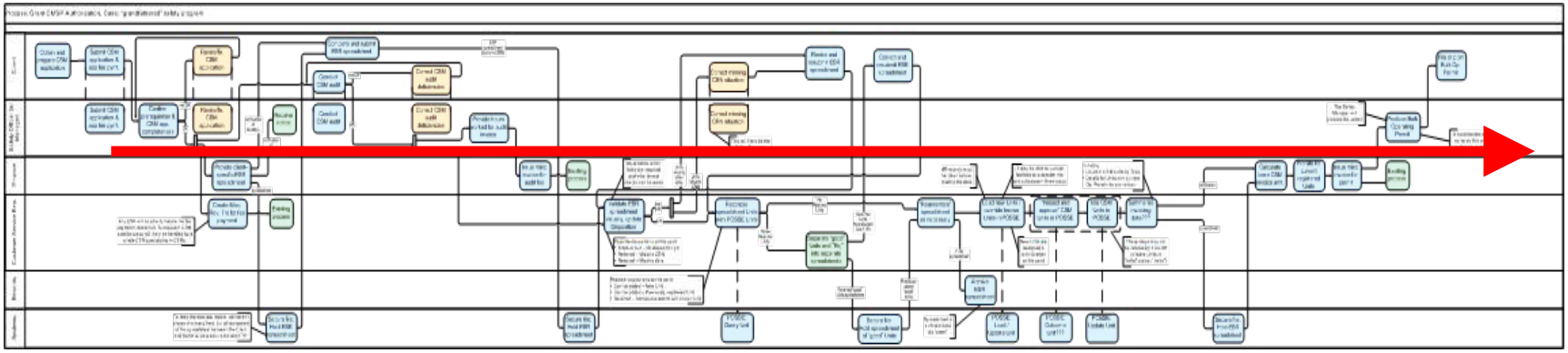
Core principles – "Flow first, detail later" and "Simplicity!"

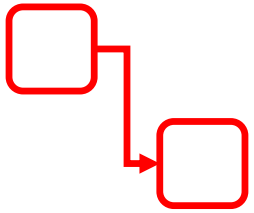
The purpose of a *Workflow* Model is to show the *Flow* of *Work*

Whatever you call them,
they are a *great* tool for showing flow –
sequence and dependency of steps

- Swimlane Diagram
- Workflow Model
- Process Map
- Cross-Functional Flowchart
- People-Process Chart
- Functional Deployment Diagram
- Process Responsibility Diagram
- LOVEM Diagram
- …

Left-to-right flow





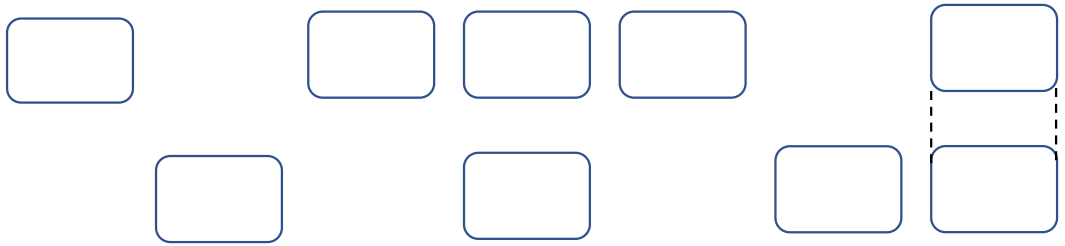
Simple… but not *simplistic*

Symbols were just boxes and lines

The Cognitive Psychology of diagramming

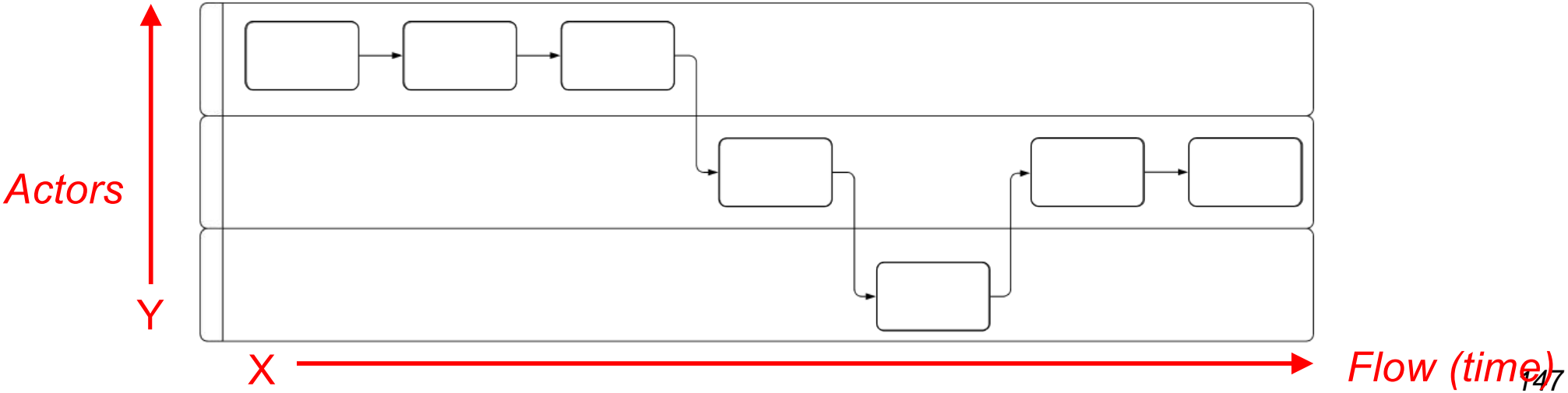
What do people first perceive on a diagram?

1. relative size

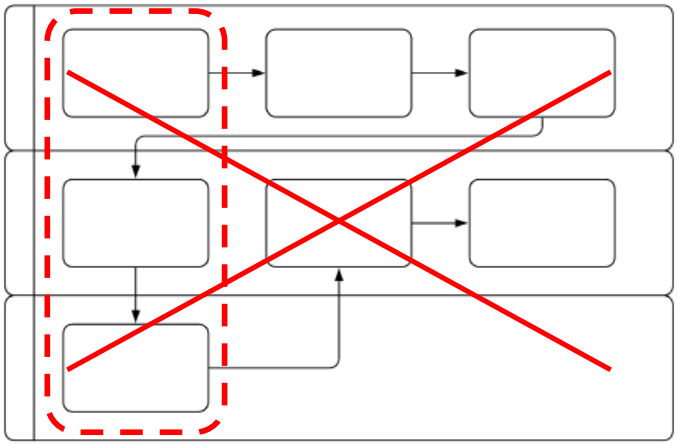


Make all the steps the same size,
unless you're trying to make a point

2. relative X-Y position



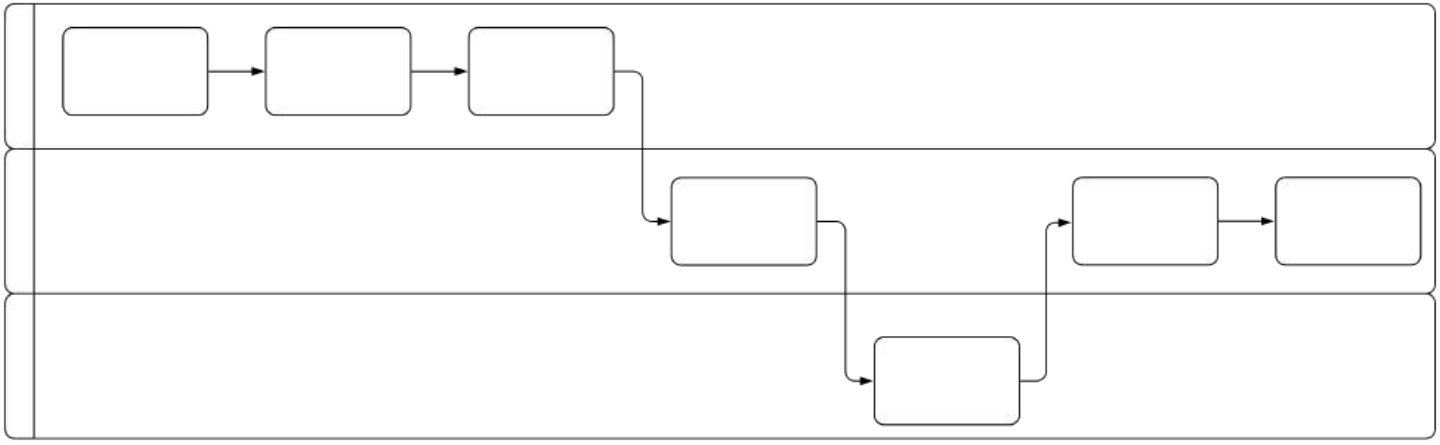
Don't conceal sequence and dependency



Steps perceived as happening in *parallel*, even though flow lines indicate *sequential*.

Critical in analysing a process:

- sequential vs. parallel
- dependent vs. independent



A simple guideline: flow lines *only* leave the right edge and *only* enter the left edge – never the top or bottom.

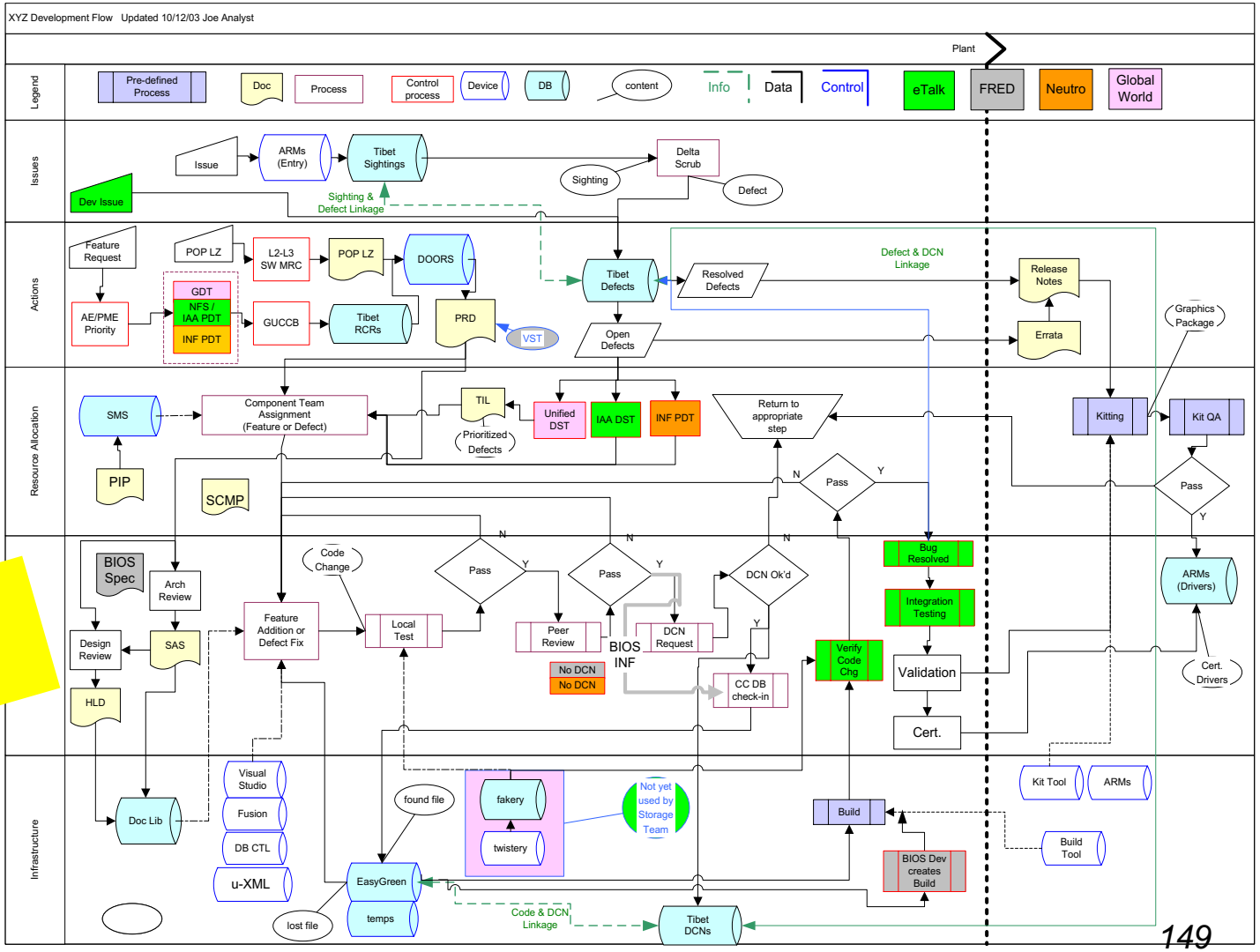
"I think I know why our business partners don't want to review this…"

Probably a good DFD (Data Flow Diagram) useful to a technical audience

Not a good Process Flow Diagram, useful to business professionals, because...

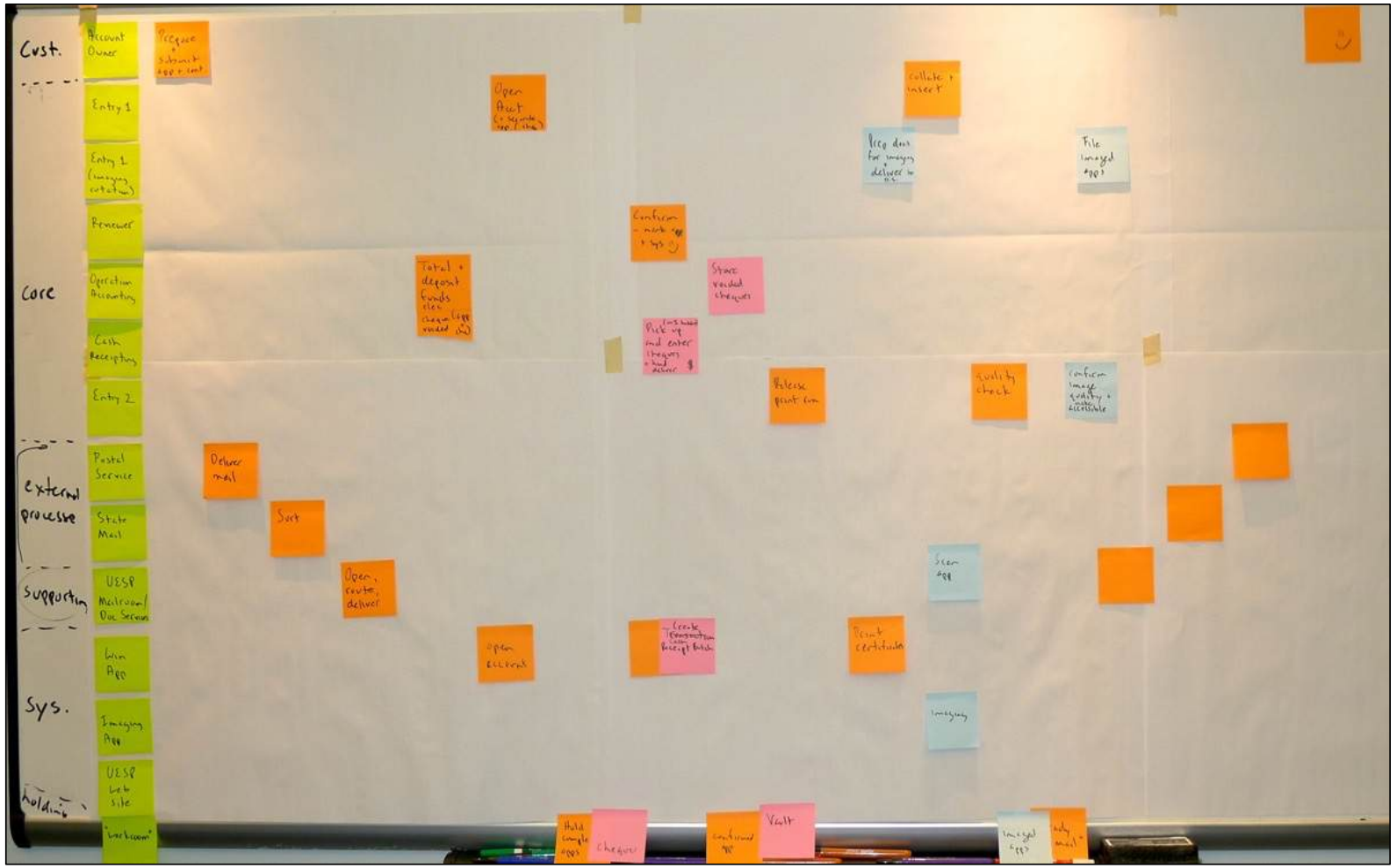
- no obvious flow
- too many symbols
- cryptic acronyms
- lanes aren't actors
- …

What's wrong with this diagram as a means of communicating with a business audience?



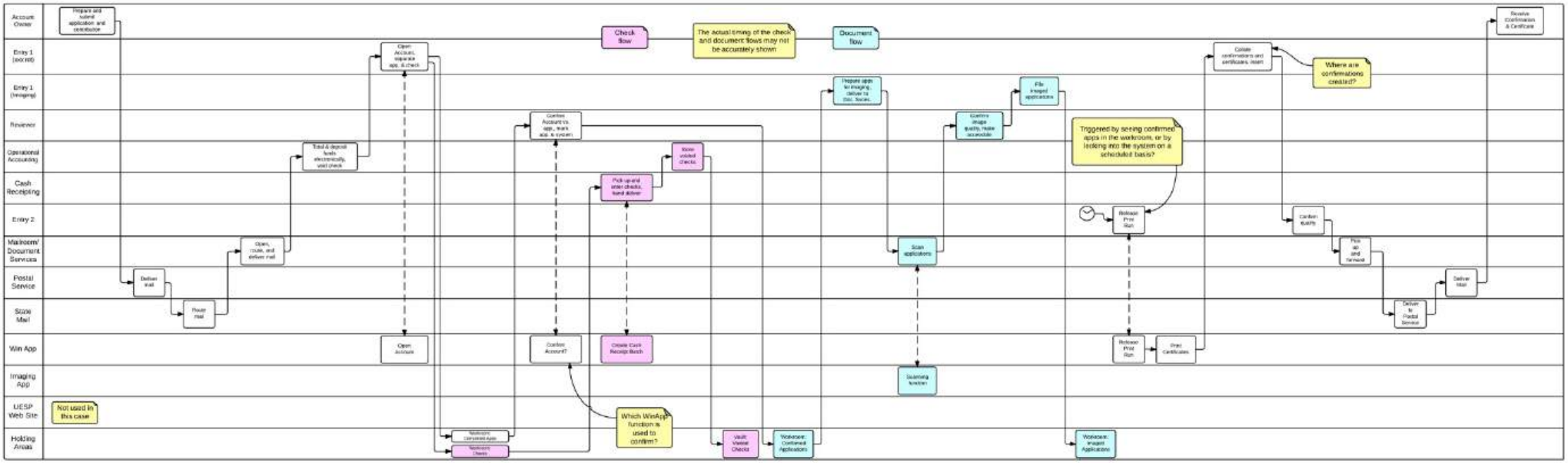
Boxes alone are a great start

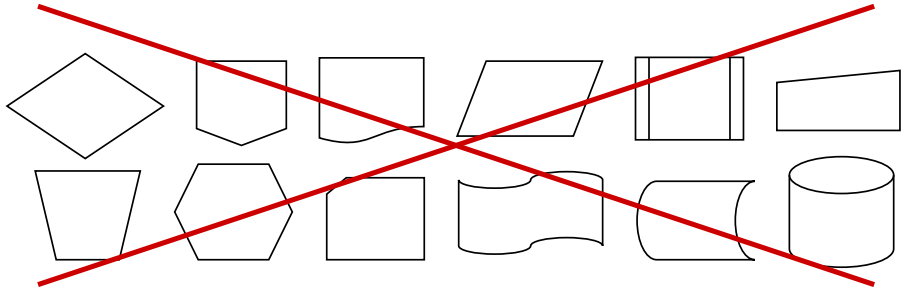
Remember – you can build an initial flow model with Post-its, real or virtual

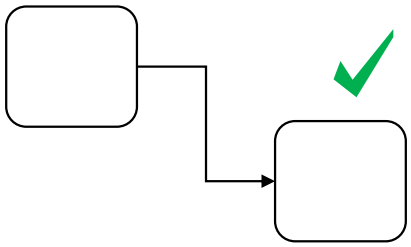


Minimal symbols

Later, redrew it with *Lucidchart* (www.lucidchart.com) – add rigour, but still focus on flow and simplicity.

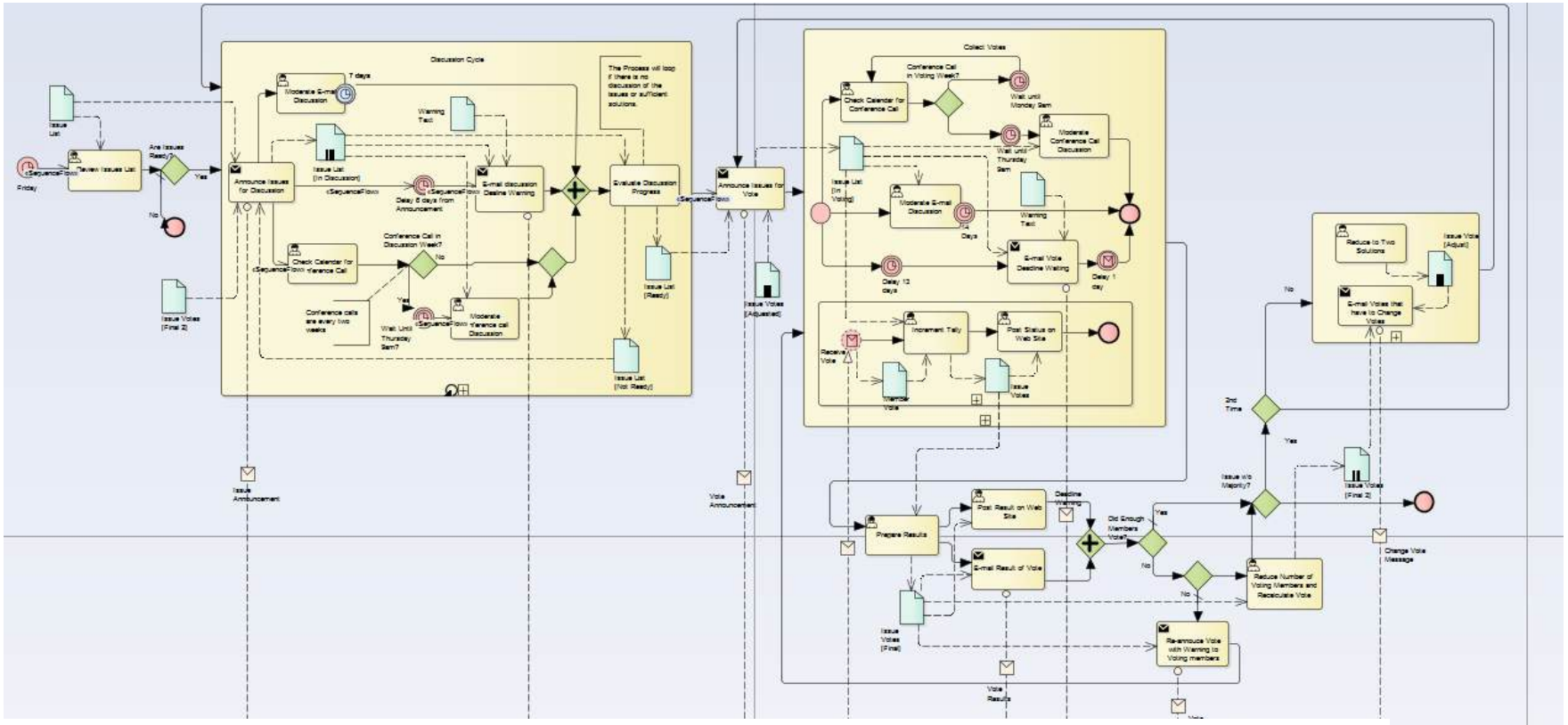






Full BPMN – not useful for business purposes*

If you choose to use BPMN, use extreme restraint! Only use a few symbols such as Lanes, Tasks, Flows, simple Events, and optionally Gateways



*Business Process Model and Notation, a standard by the OMG. BPMN was created to be a *visual programming language* for automating workflow, not for business modelling.

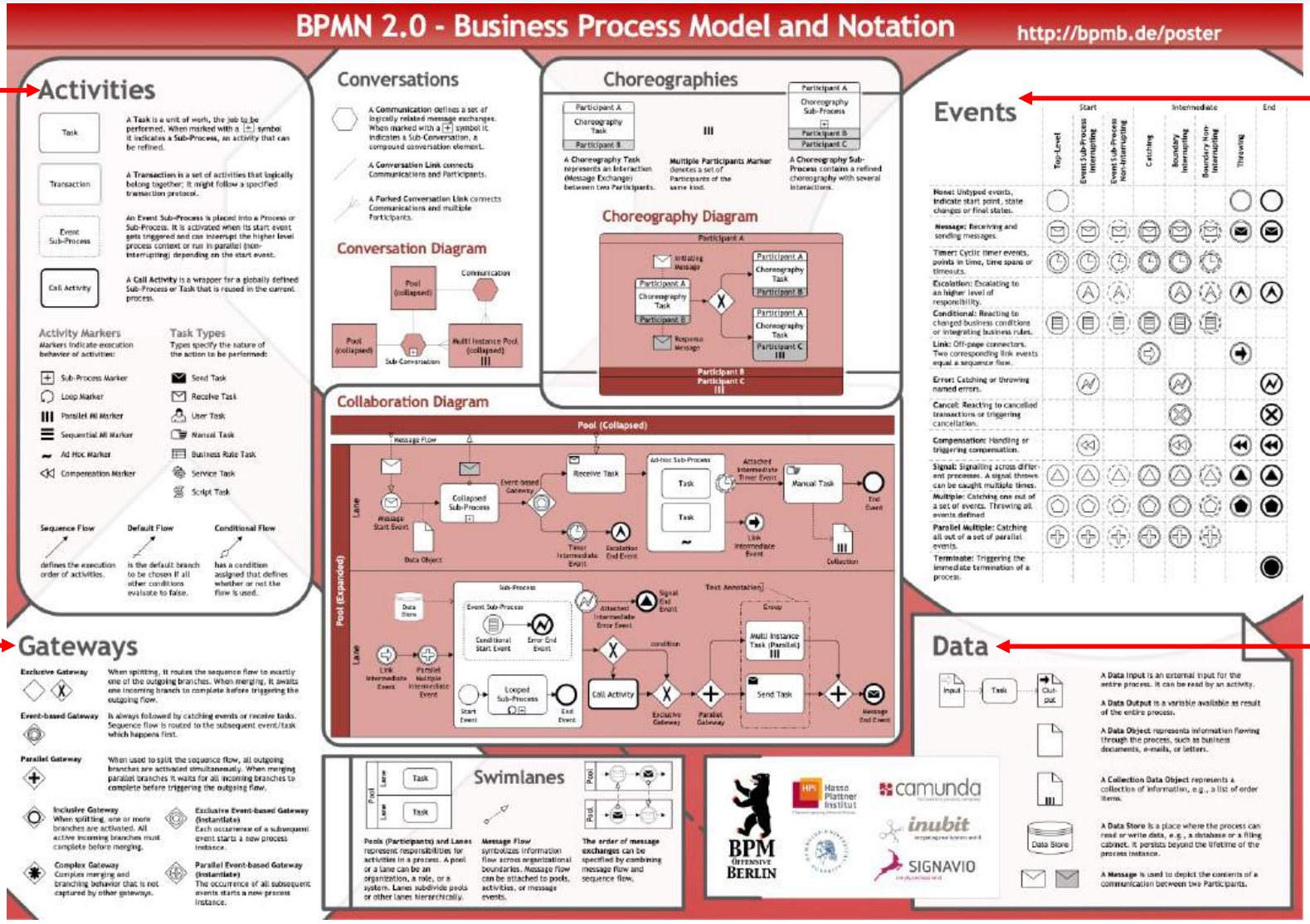
The full BPMN symbol set (why we use a subset)

4 kinds of Activities (plus Markers & Task Types)

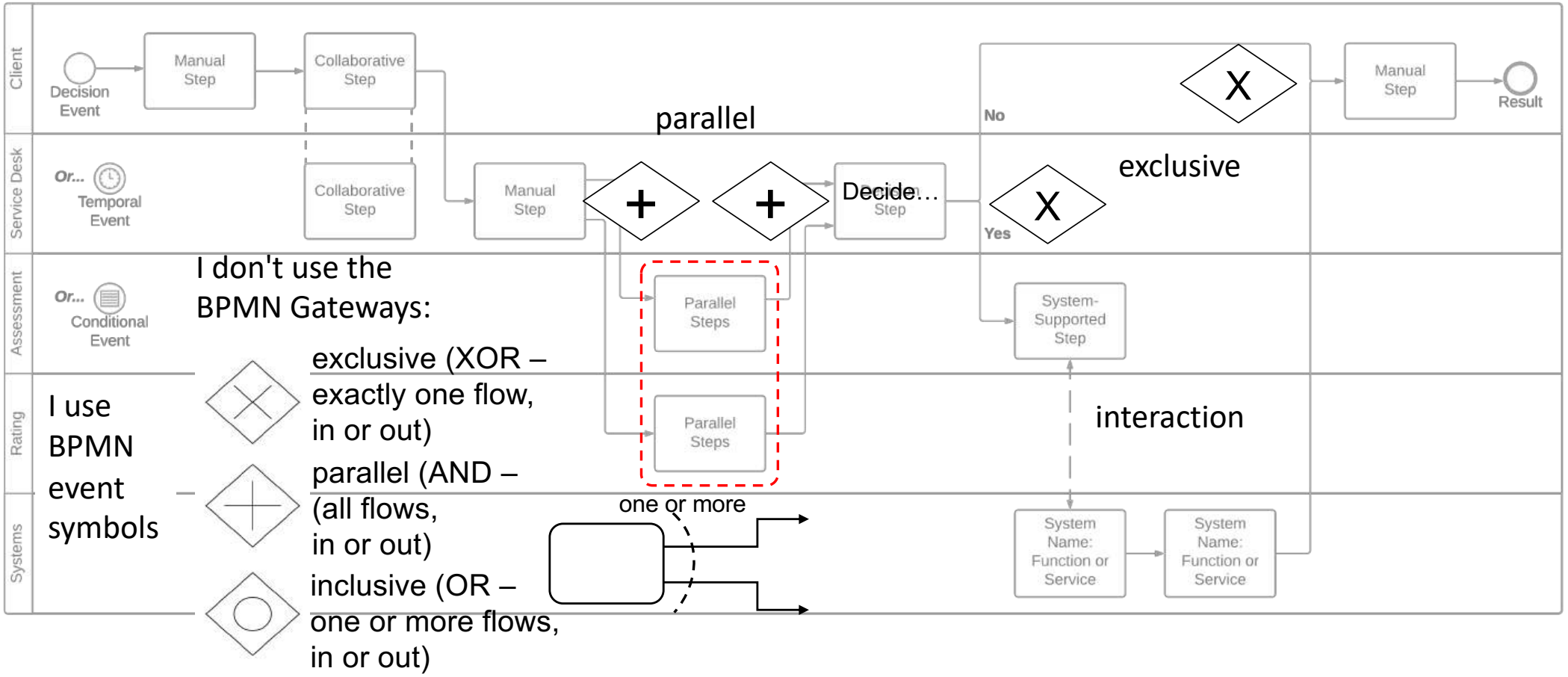
63 kinds of Events

7 kinds of Gateways

6 ways to represent Data



Minimal symbols for an approachable workflow model



Three questions to develop your initial workflow model

Emphasis:

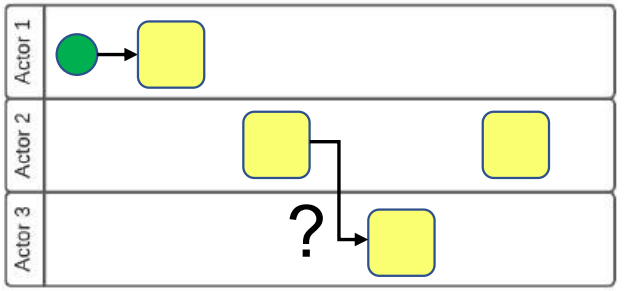
- keep you out of the details – focus on *flow*
- ensure the involvement of *every* actor is shown – it doesn't matter *how much* or *how little* they do, or whether they *add value*

Three simple questions:

1. "*Who* gets the work next?"

2. "*How* does it get there?"
   – Often uncovers "transport" actors or systems

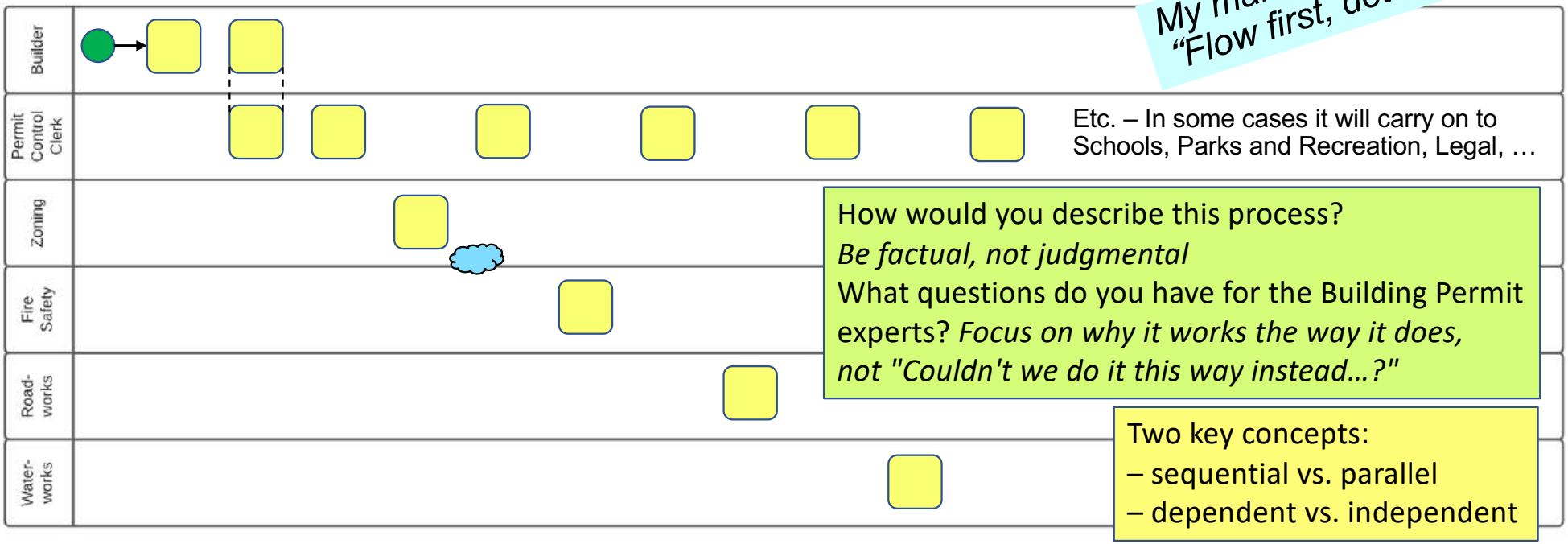
3. "Who *really* gets the work next?"
   – Often uncovers additional actors



Guideline for the initial Handoff Diagram:
Whenever an actor *holds the work*, whether they do a *lot* or a *little*, draw *one* box (or post *one* sticky) and *move on!*
(And no value judgements – include *every* actor that holds the work!)

Question 1 – "Who gets it next?" traces overall flow

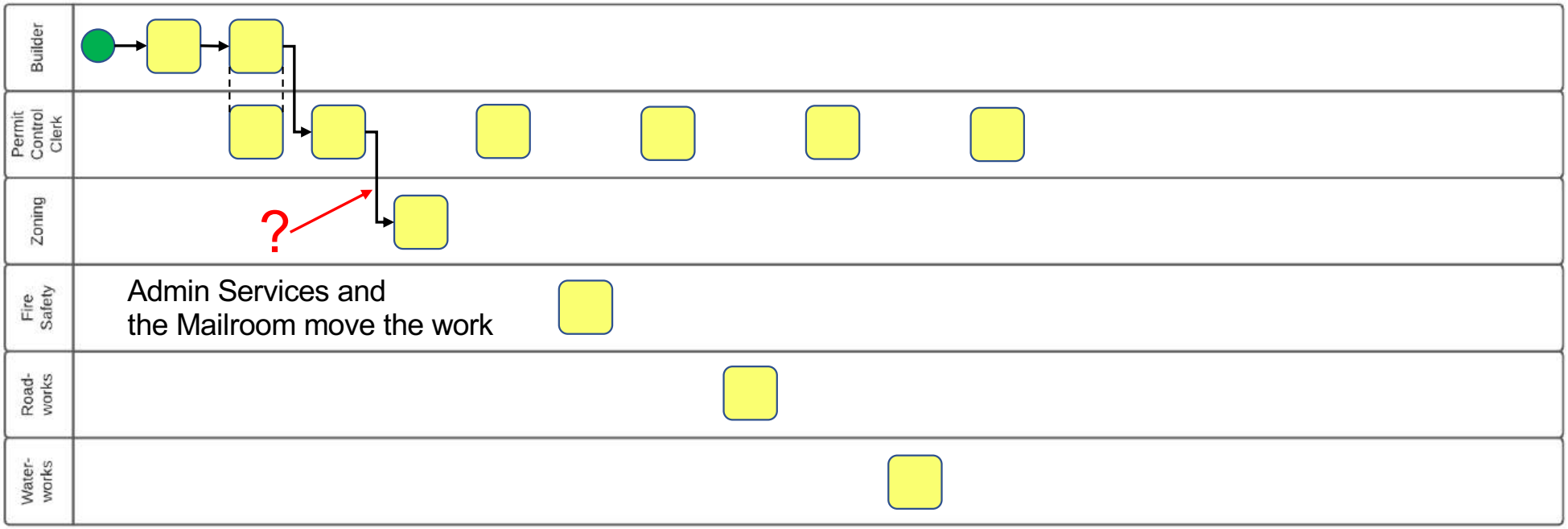
Process: Issue Building Permit
Case: Single Family Dwelling (SFD)



In scoping, you identified the trigger, the result, and the main actors. Now, starting at the triggering event, keep asking question 1 –
"Who gets the work next?"
- trace the flow of work through to the Customer's result, following one path only!
- at a decision or parallel flows, follow the main path, mark the other with a cloud, and return later
- ***DO NOT*** ask "What do you do?"

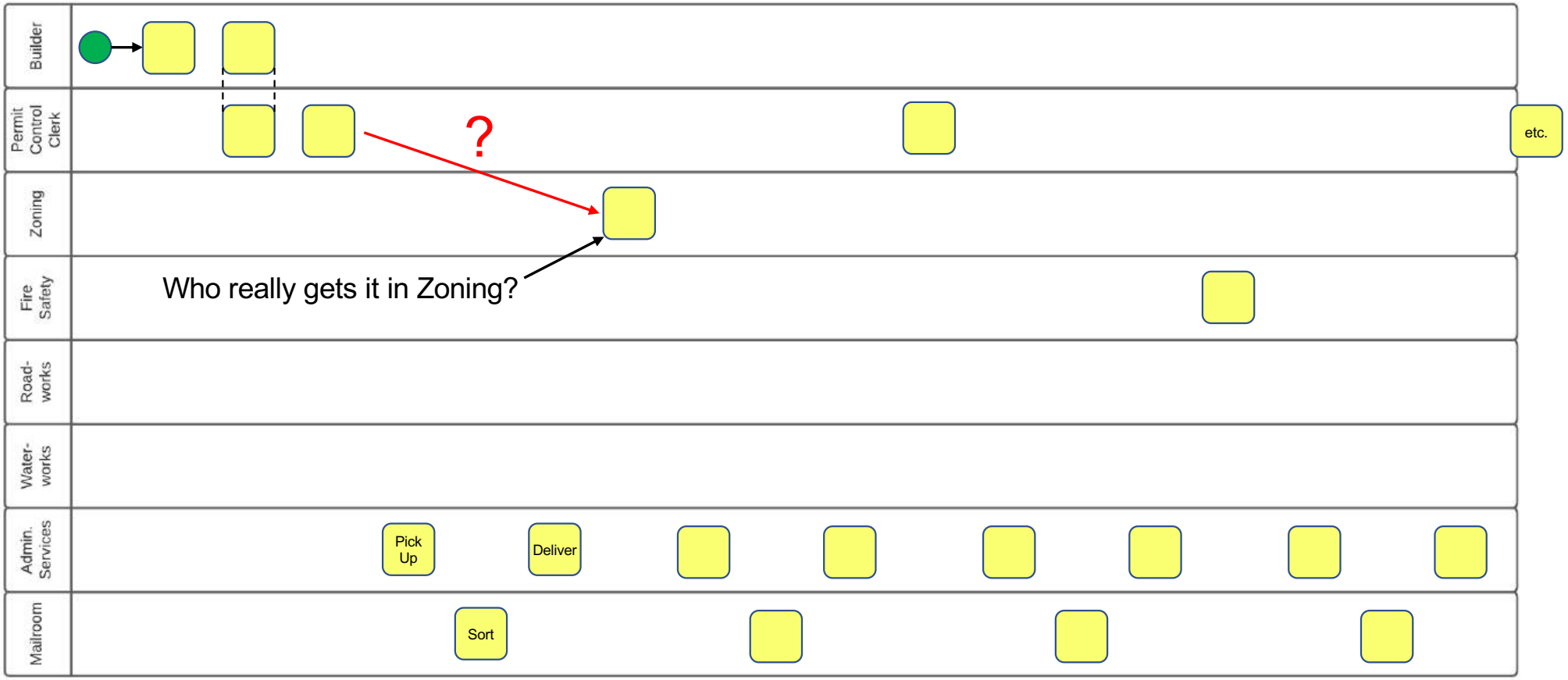
Question 2 – "How does it get there?" uncovers more actors

Process: Issue Building Permit
Case: Single Family Dwelling (SFD)



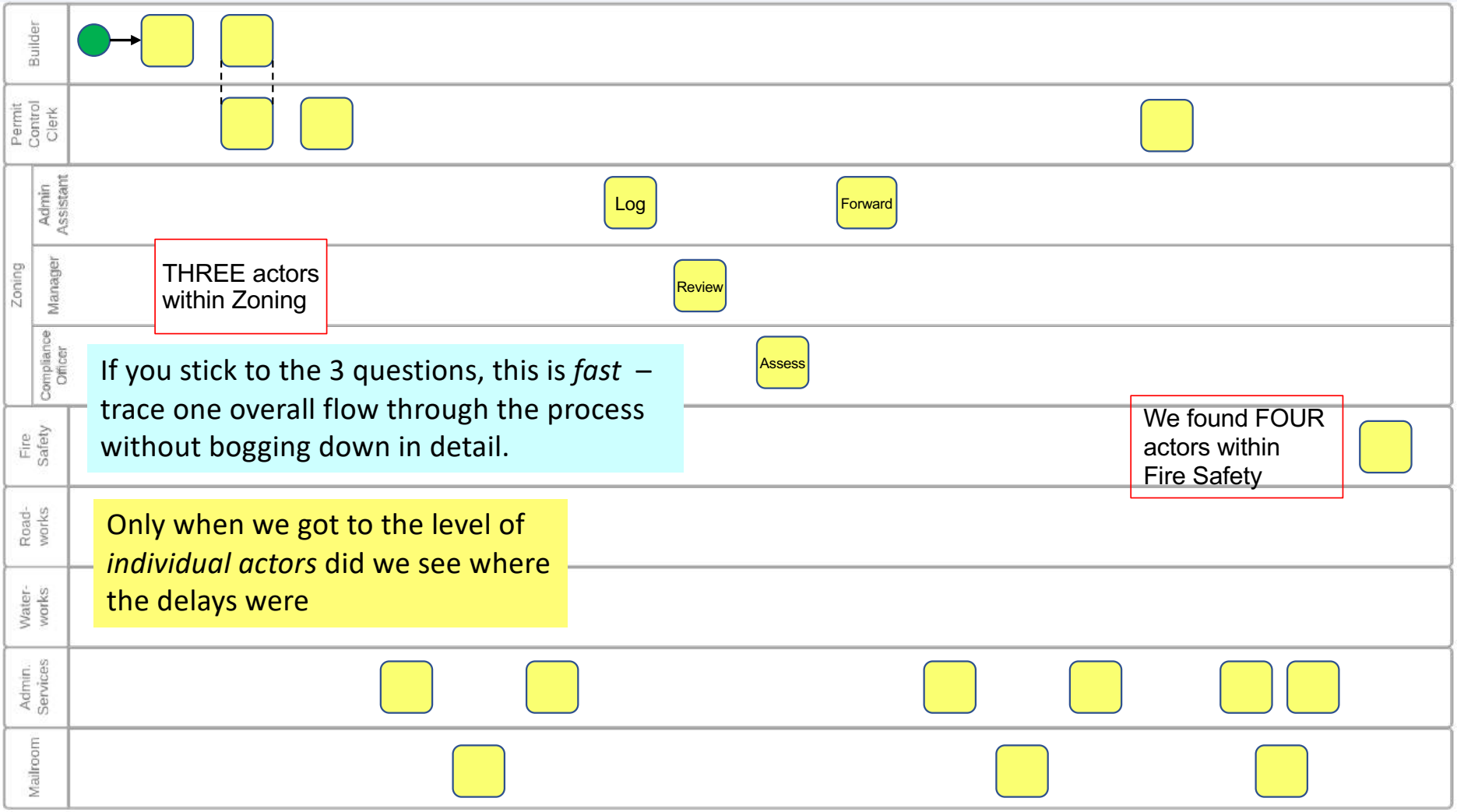
Next, at every handoff, ask question 2 –
"How does it get there?"
- uncovers *additional actors*, and therefore more handoffs
- a handoff is a potential source of *delay, error, or expense*

Question 2 revealed more actors and transport mechanisms

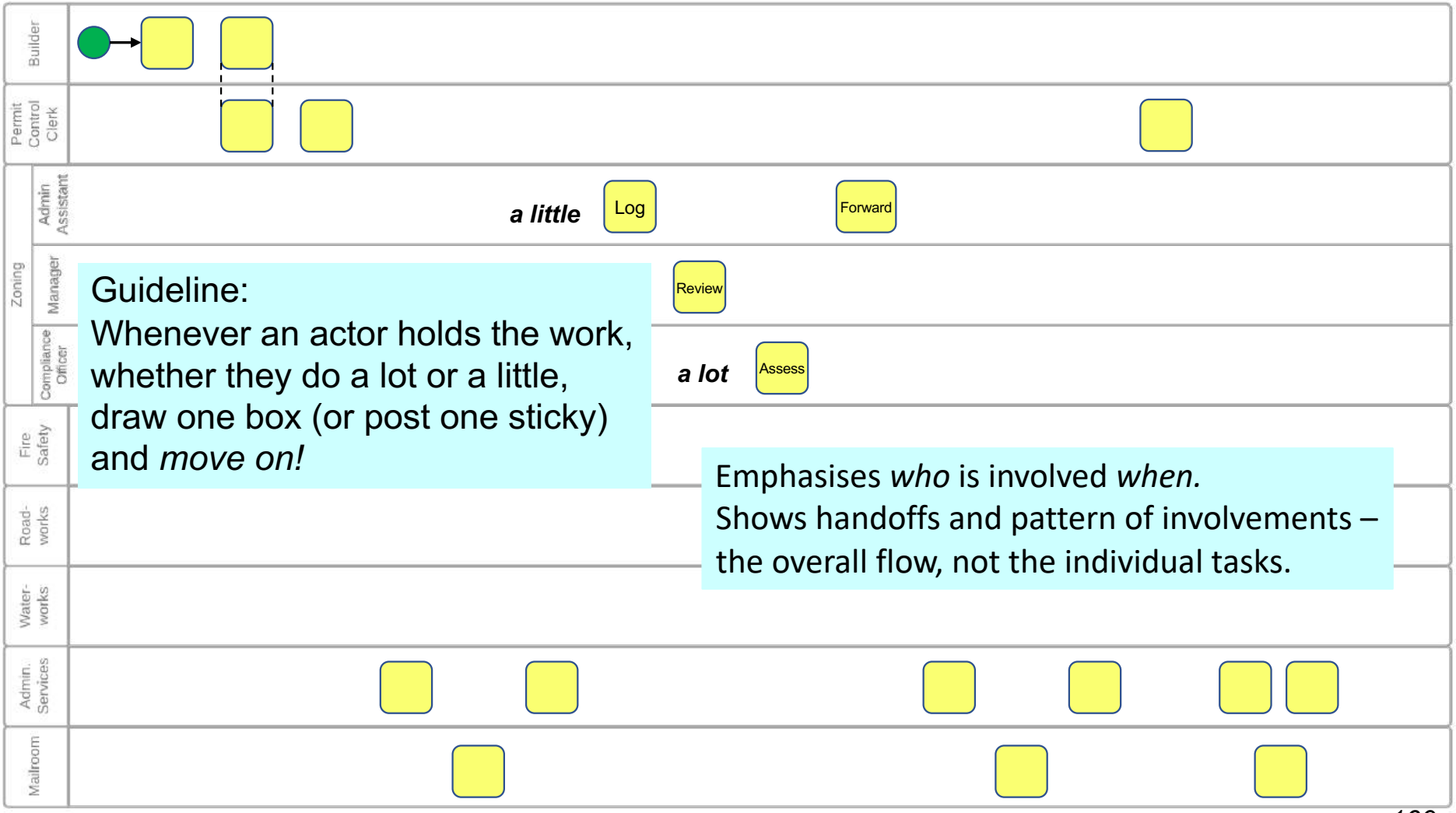


Now, inspect handoffs again, looking for missing actors, ask question 3 -
"Who really gets it next?"
- does it *really* go directly to the actor you first identified?

Question 3 – "Who really gets it next?" uncovers specific roles



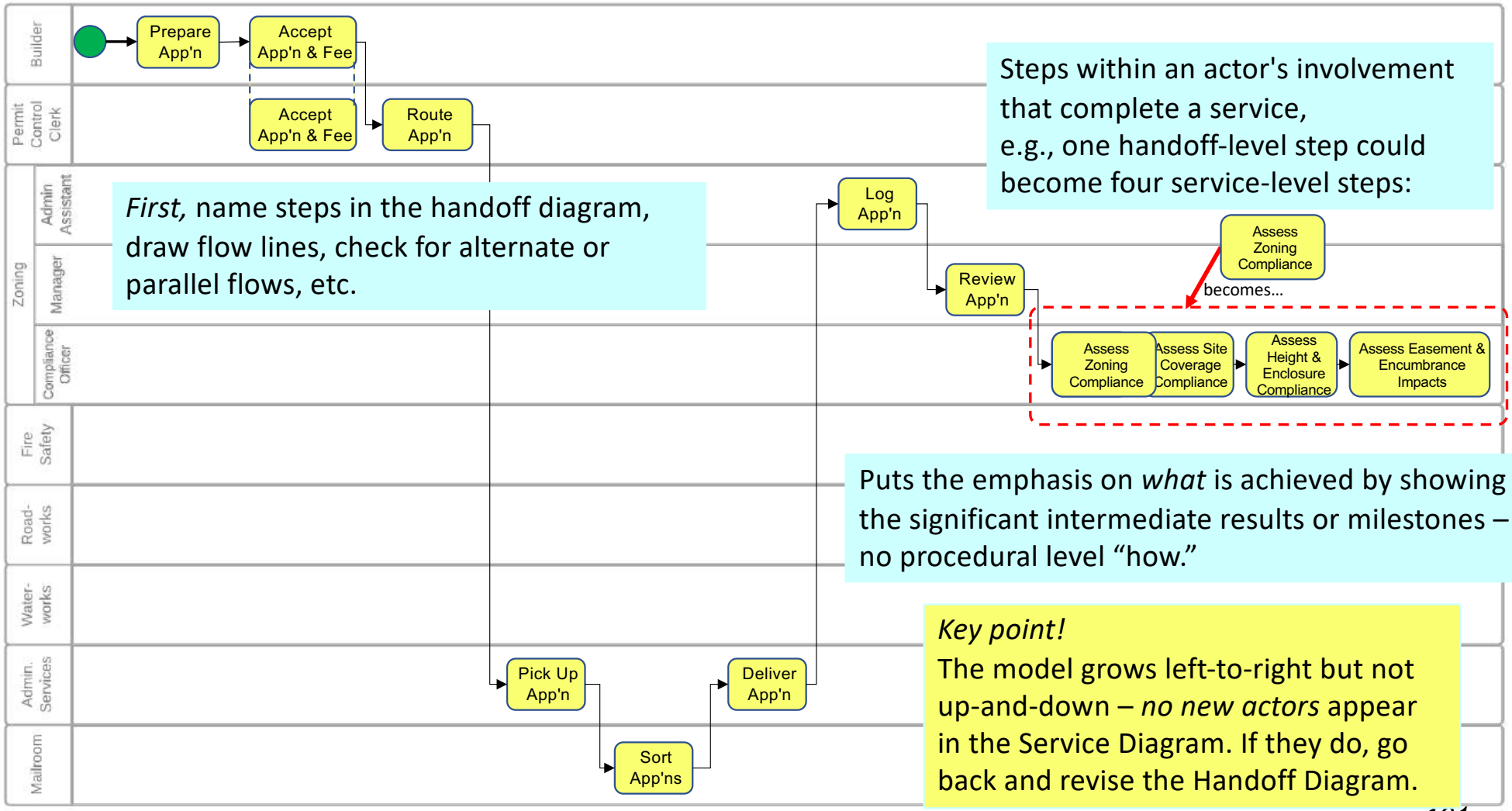
We have started a "Handoff Diagram"



Guideline:
Whenever an actor holds the work, whether they do a lot or a little, draw one box (or post one sticky) and *move on!*

Emphasises *who* is involved *when.*
Shows handoffs and pattern of involvements – the overall flow, not the individual tasks.

Now develop a "Service Diagram"

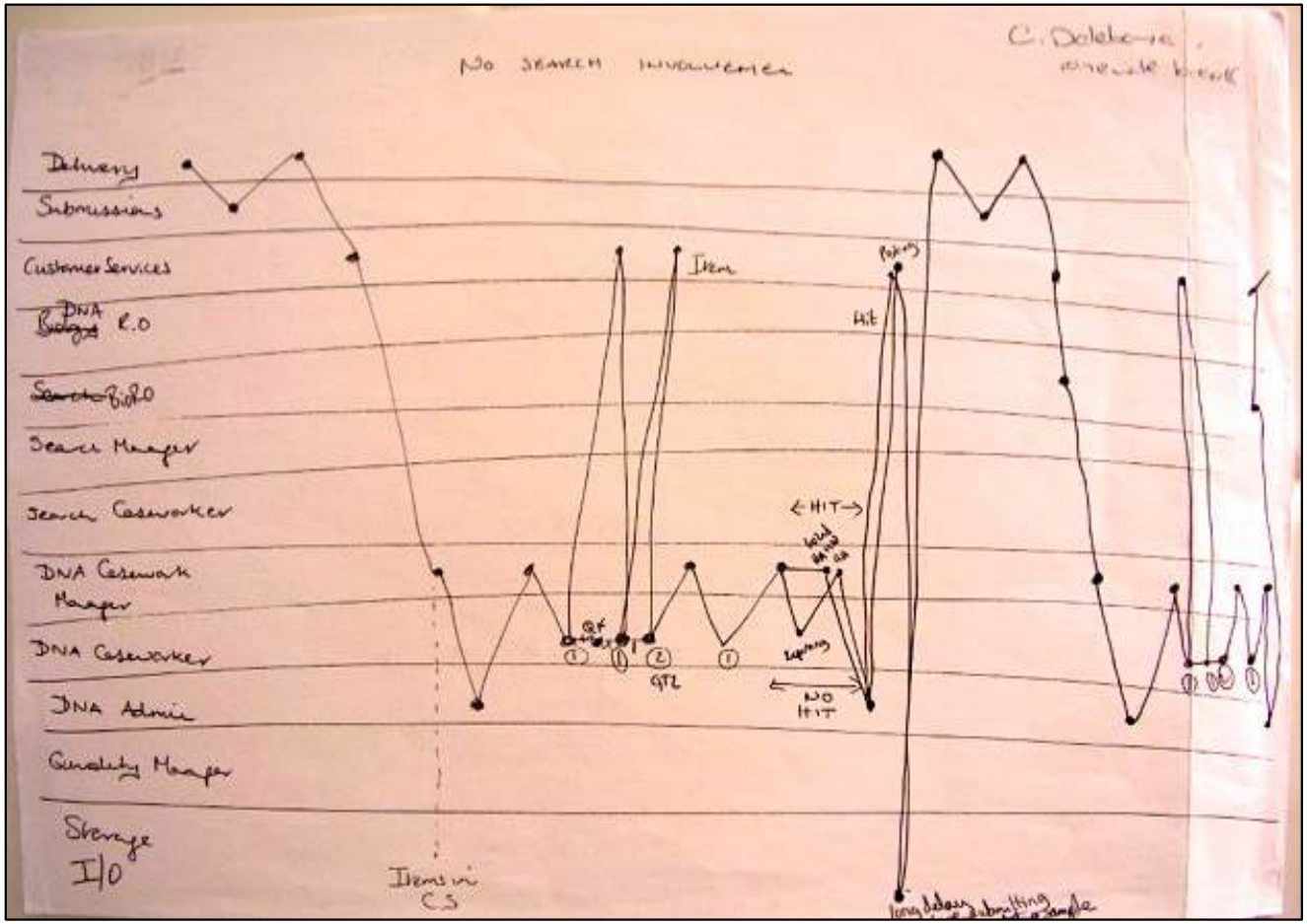


Two levels of swimlane diagrams

Level	Definition	Emphasis	Notes
1 Hand-off	• Draw one step (box) every time an actor continuously "holds the work," no matter how much or little they do	"Who"and "When" - pattern of involvement	• Sometimes this level of detail is enough to understand As-Is process behaviour
2 Service	• Decompose handoff-level steps into discrete services, *as necessary:* one step each time actor achieves a significant result or state change	"What" is actually achieved	• Usually, we don't go any further than this for the As-Is process • Also called a "Milestone" diagram

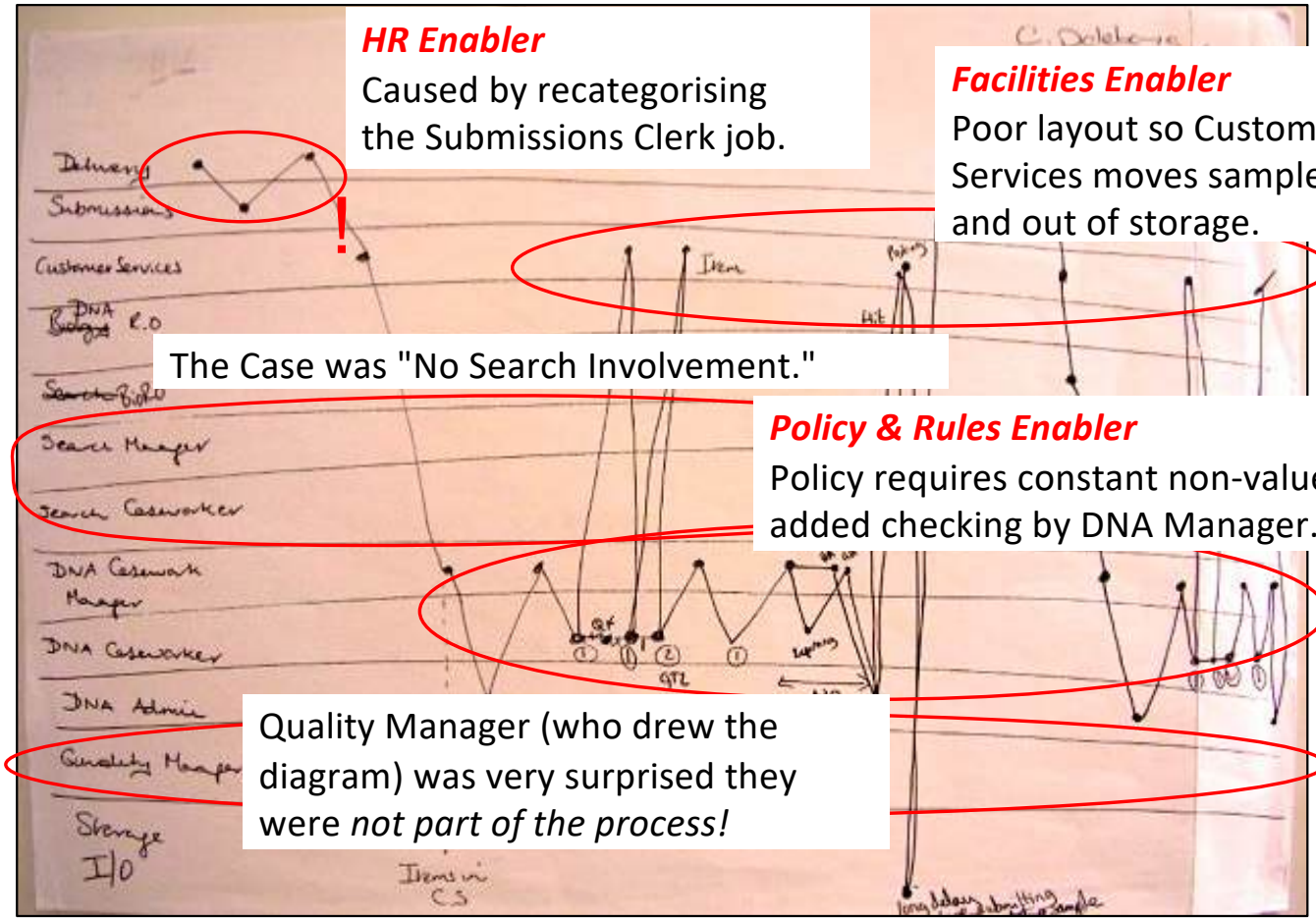
The handoff-level diagram is critical – ensures we discover the overall flow before diving into detail.

What stands out on this minimalist Handoff Diagram?



"Order and simplification are the first steps to mastery of a subject."
Thomas Mann

We learned a LOT in a short period of time



HR Enabler

Caused by recategorising the Submissions Clerk job.

Facilities Enabler

Poor layout so Customer Services moves sample in and out of storage.

The Case was "No Search Involvement."

Policy & Rules Enabler

Policy requires constant non-value-added checking by DNA Manager.

Quality Manager (who drew the diagram) was very surprised they were *not part of the process!*

Business Process Design (Workflow)	Technology & Information Systems	***Motivation & Measurement***	***Human Resources & Organisation***	***Policies & Rules***	Facilities (or, **Knowledge / Info / Data**, Communications, Documents, …)

The Service level workflow

- Purpose -

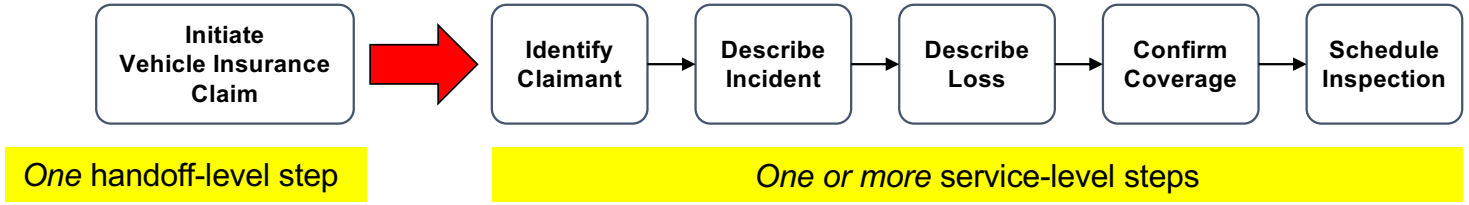
Understand the actual contribution of each actor to the process

Ensure feasibility and effectiveness of process (can each actor actually perform their steps?)

Show relationship to systems - steps involving automated support correspond strongly to use cases and services

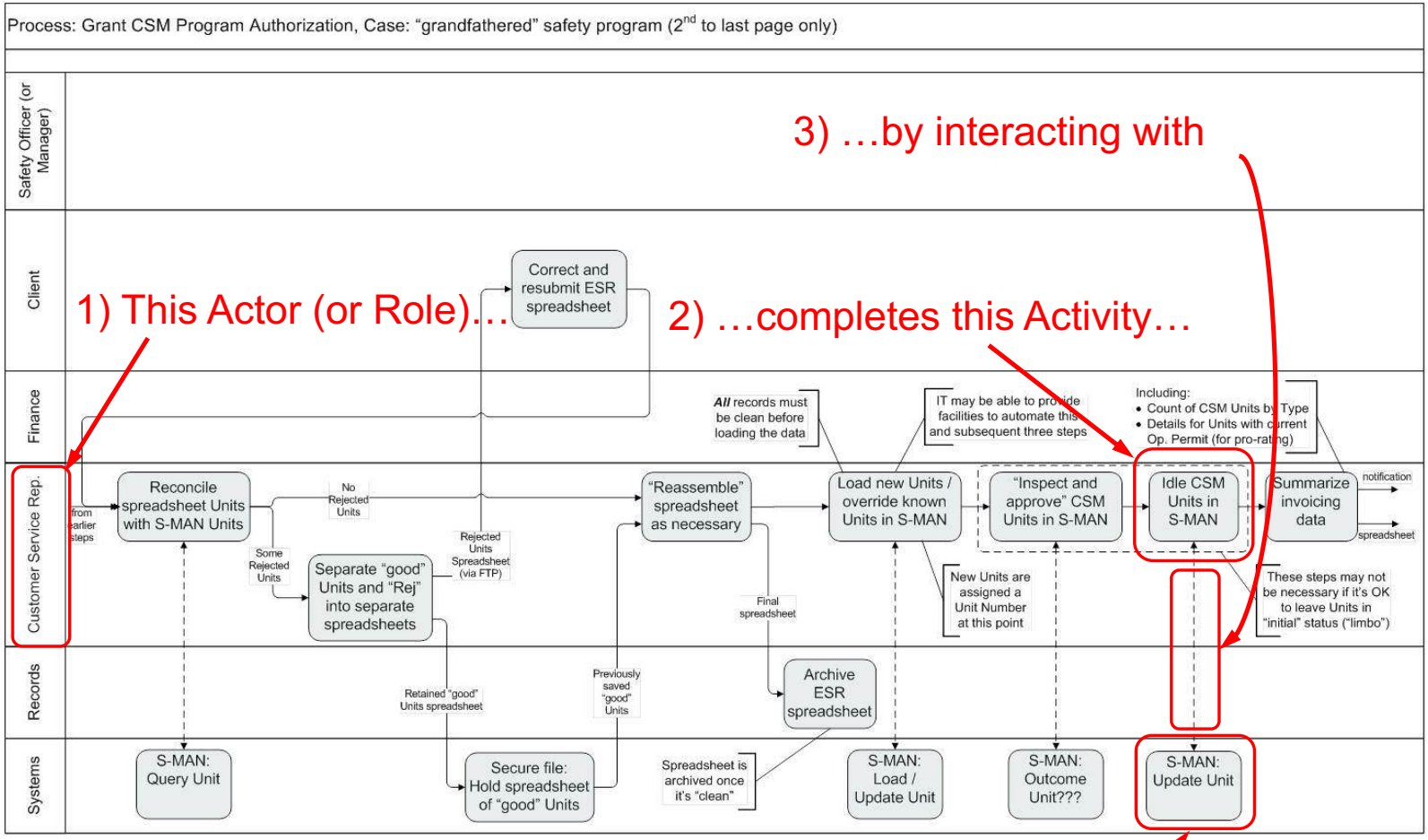
Key points:

- Steps within an actor's involvement that complete a service
  E.g., *one* handoff-level step could become *five* service-level steps:



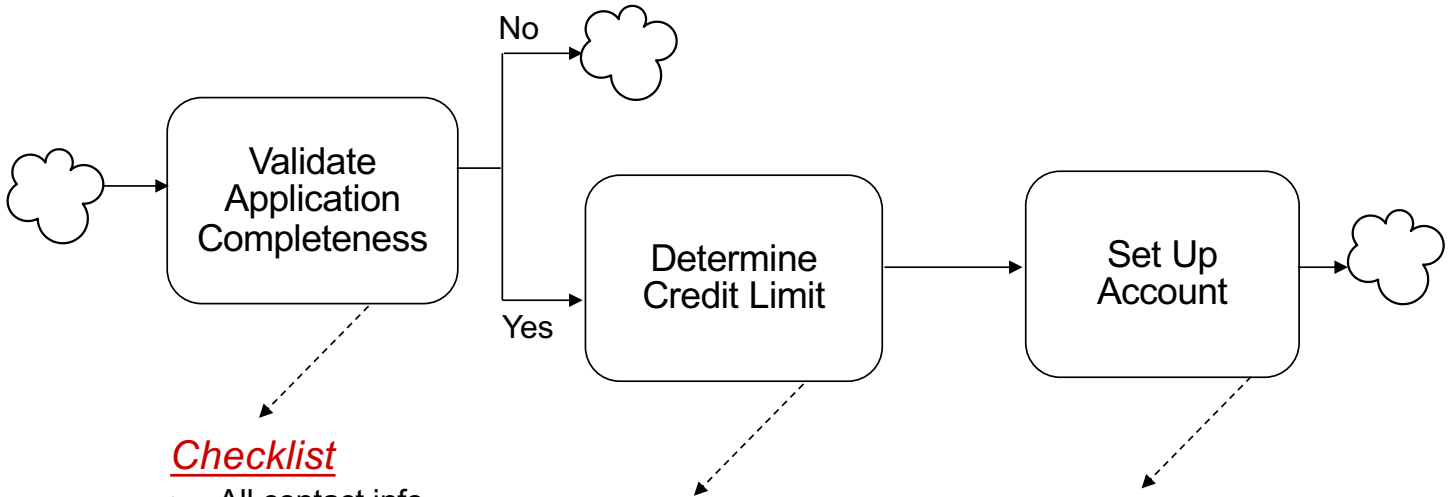
- Puts the emphasis on *what* is achieved during the process by showing the significant intermediate results or milestones –
  "the achievements, not the individual tasks"

Reminder: the service level ties in Use Cases and Services



Stop diagramming before you get into "how"

Stop workflow modelling when work isn't flowing.
Do ***not*** use a workflow model to describe ***how*** an activity is done – that belongs in the activity description or in a linked document.



Checklist

- All contact info complete?
- Tax registration number provided?
- Net income <= gross income?
- etc.

Procedure

- Determine gross annual income
- Subtract annual tax
- Subtract child support
- Etc.

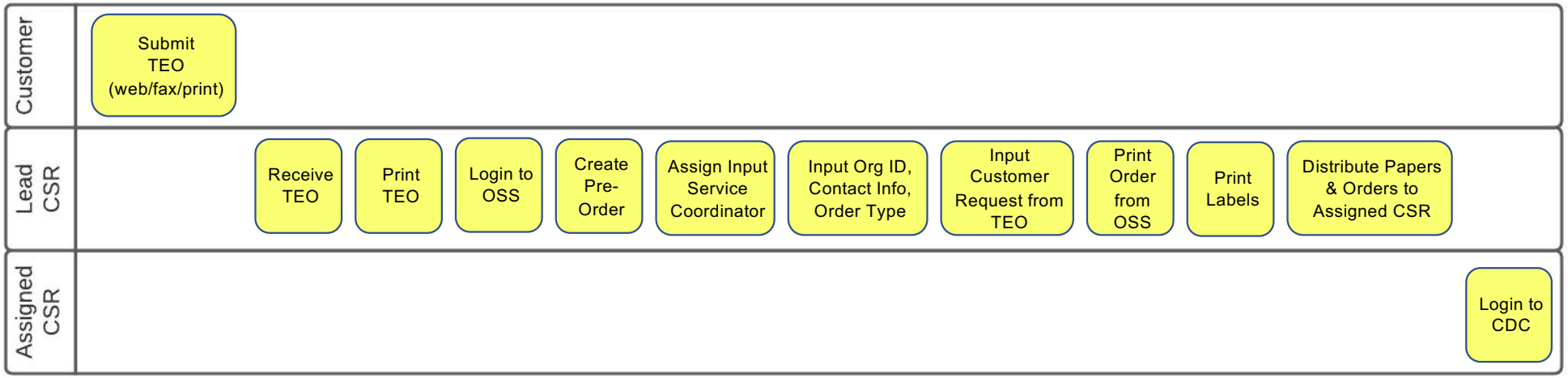
Use Case

- When credit rep enters… Then system displays…
- When credit rep selects… Then system calculates…
- etc.

Knowing when you've gone too far

Do ***not*** use a workflow model to describe ***how*** an activity is done – that belongs in the activity description or in a linked document.

Handle TEO
(Telecom Equipment Order)

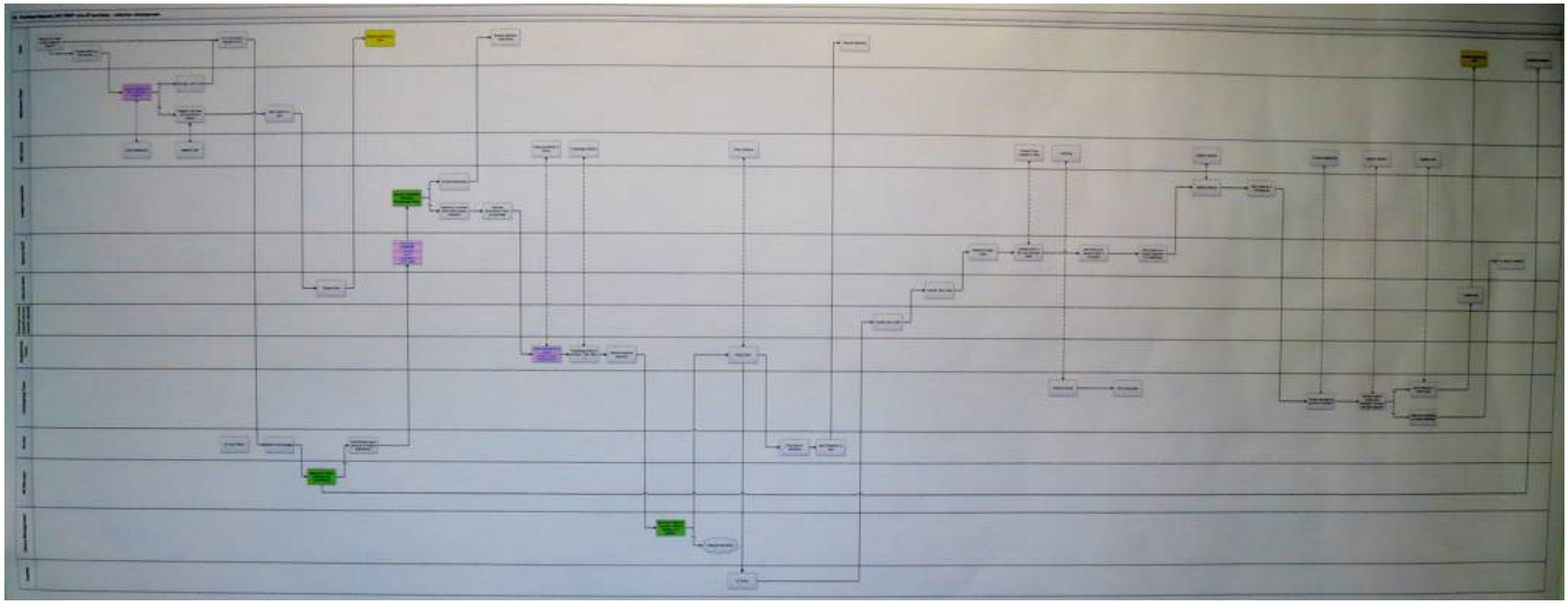


You've gone too far if:
- there are multiple steps in sequence by the same actor
- the steps include "how-to" instructions (procedural level detail)

Getting out of the weeds if you've gone too far

Client struggling with process redesign, mired in detail

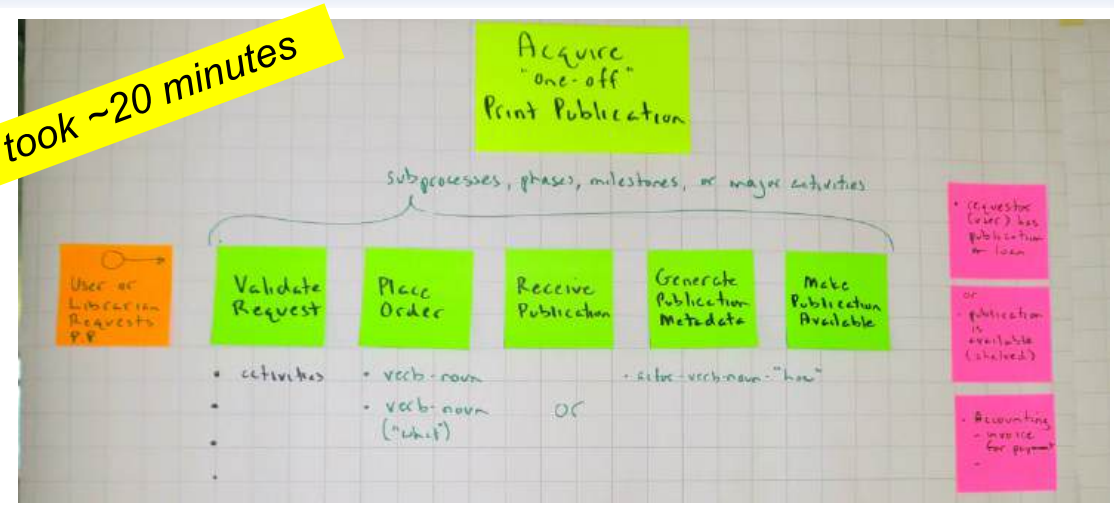
One of 17 flow models for variations of the same process



Excellent models, but detail without context is the enemy!

Step One – Establish context with a Process Scope Model

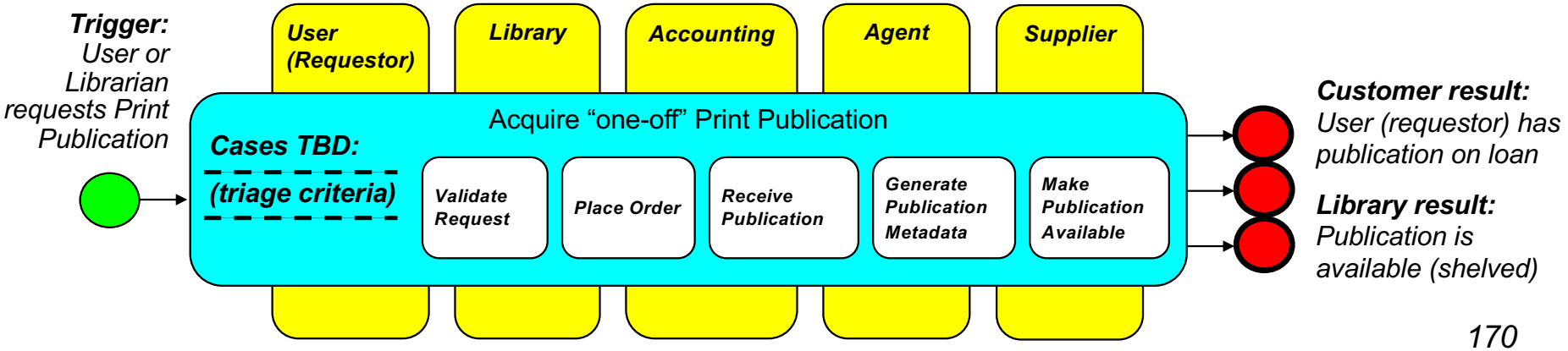
Step 1 – build a Process Scope Model then a Summary Chart



The process was actually "Acquire Information Asset" and the Case was "One-off Print Publication"

Cleaner version using our "TRAC" framework – Trigger, Results, Activities, Cases:

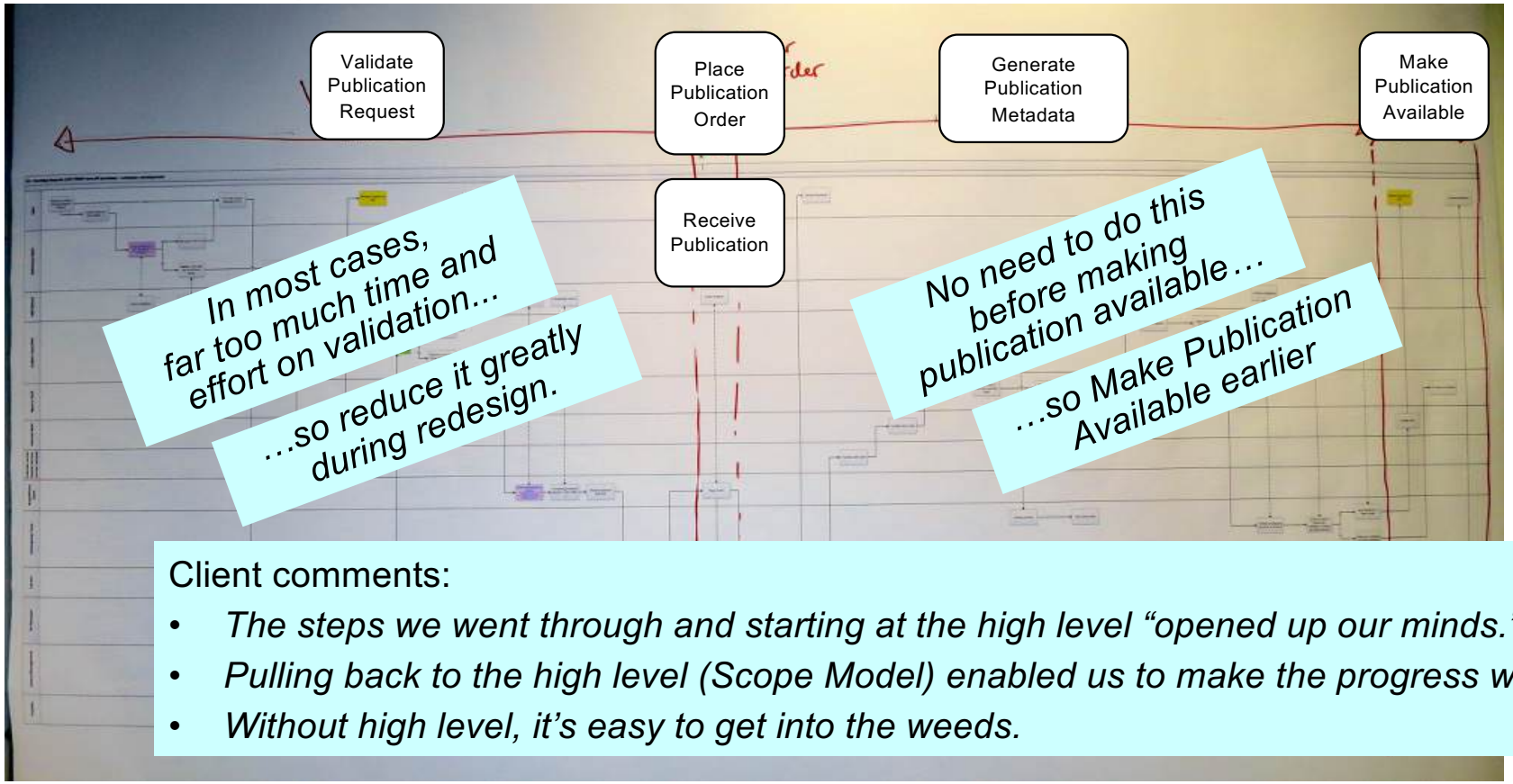
Then add participating functions:



Process Scope Model puts the detail into context

One of the clients had a great idea –
overlay the phases from the *Process Scope Model* on the *Workflow Model.*

90+ % of the process is activities the customer dislikes or doesn't care about!



Client comments:

- *The steps we went through and starting at the high level "opened up our minds."*
- *Pulling back to the high level (Scope Model) enabled us to make the progress we did.*
- *Without high level, it's easy to get into the weeds.*

Summary – where we've been, where we're going

Principles

The purpose of a *Workflow* Model is to show the *Flow* of *Work*

Simplicity is a virtue

Always do a Scope Model and a Summary Chart before flow modelling

Why they work

Flow (sequence & dependency) is clearly visible, left to right

Simple to read – the symbols are mostly boxes and lines

Shows all actors and their steps, and therefore all interactions and handoffs

Shows the entire, end-to-end process, from trigger to results

Shows "what" the steps are without diving into "how"

The most *common* errors

Concealing flow by drawing a convoluted diagram, usually in an attempt to make it a "one-pager"

Using a lot of symbols that regular folks don't understand

Omitting actors just because they play a minor part – *everyone* has an impact

Cutting the diagram into one-page segments – the initial flow model should be continuous

Using a Workflow Model to document procedural level detail

Our methodology – two points highlighted by clients

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***

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

3 Design the To-Be Process

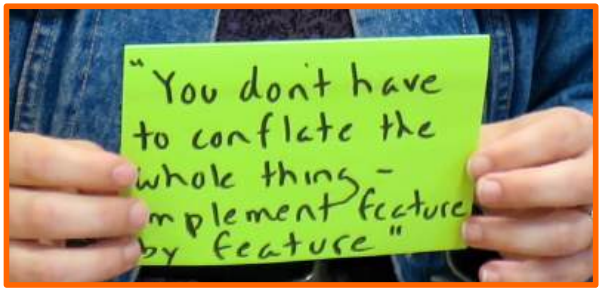
- Refine to-be improvement ideas and determine 5-10 ***key features*** of the to-be process
- Assess each to-be feature ***by enabler*** to ensure the new process is implementable and sustainable
- Design the to-be process:
  1 - ***essential*** activities first
  2 - "who & how" next
  3 – transport & protocol last

Start with *what* | *Inclusive* assessment | Based on *reality* | Awareness of *all* factors | Addresses *our* goals | We can *do* it! | Feature-based approach

1 – Builds support for *change*

"We like the way support for change is built in *throughout* your approach, not bolted on at the end."

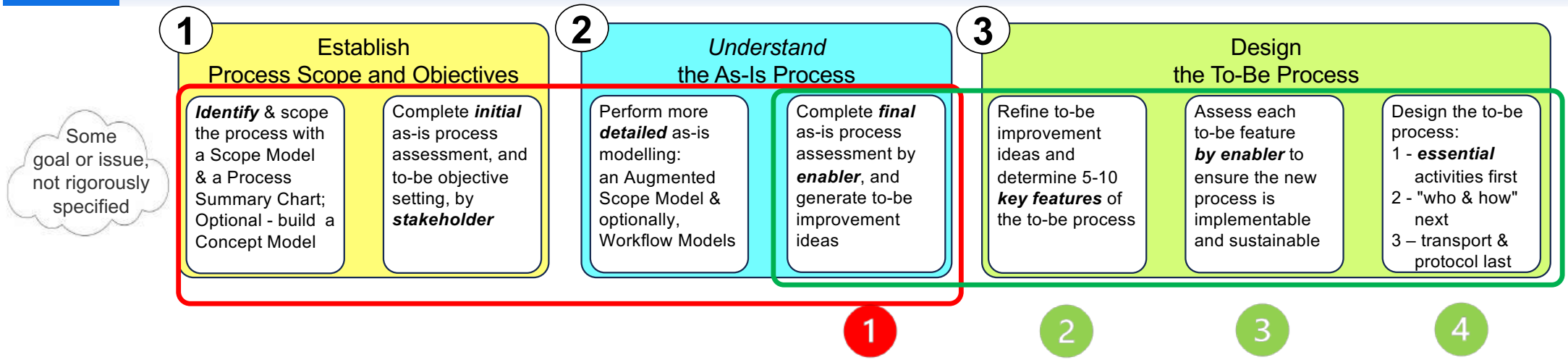
2 – Not a "big bang" – an effective, implementable, sustainable business process



Feature-based approach makes it *Agile / iterative.*

And *fast!* – up-front work avoids endless rehashing later

The link between the As-is Process and the To-be Process



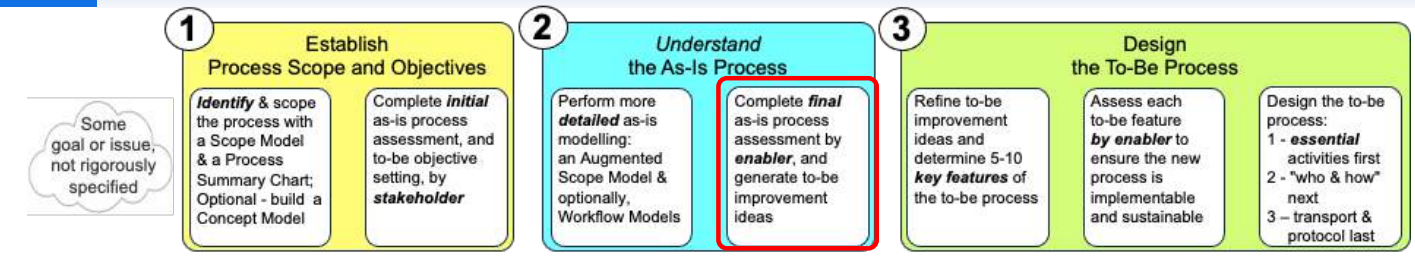
This activity (1) marks the pivot from as-is to to-be:

- we capture what we learned while studying the *as-is*
- we use this to generate ideas for the *to-be*
- three more activities (2 3 4) lead us to a new design

Key point!

Much of what we learn comes from discussions along the way, not from studying the swimlane diagram.

Complete final as-is assessment, generate to-be ideas



Considering all six enablers is at the heart of this methodology

Business Process

enables

enables

Business Process Design (Workflow)	Technology & Information Systems	***Motivation & Measurement***	***Human Resources & Organisation***	***Policies & Rules***	Facilities (or, **Knowledge / Info / Data**, Communications, Documents, …)
• Too many actors • Non-value - added steps • Duplicate steps • Delays and bottlenecks • Excessively sequential	• Unavailable information • Data re-entry • Missing functionality • Awkward interfaces • Lack of support for workflow	• Inappropriate performer or process measures • Internal rather than customer focus • Measures of tasks vs. outcomes	• Mismatches between task value and performer • Too little empowerment • Fragmented jobs / roles • Inappropriate recruiting and placement	• Out-of-date policies or numerical limits • Excessive review or approval steps • "Anecdotal" or inconsistent policies • Misinterpreted regulations	• Mismatch of work needs and facility • No support for team work • Layout that impedes flow of people or material

This *always* uncovers issues that would have been missed otherwise and *always* generates ideas (potential *features*) for the to-be process

A few examples…

Workflow AND Technology

- Failing to rethink process design to take advantage of new technology…
- *The new "Settle Claim" process was still completely sequential after implementing a Workflow system because they copied the old paper-based workflow*

Motivation and Measurement

- What you measure is what you get…
- *Customer Service Representatives: measured on not exceeding 2 minute call time, so they hung up on Customers at 1:58 or 1:59*

Human Resources

- Depressingly common…
- *Clerical, administrative, and support staff made redundant, so highly-paid professional staff do the work instead (and poorly)*

Policies & Rules

- Micromanagement…
- *Laboratory technicians: work had to be checked by a senior manager after every step, so the process was bogged down in pointless reviews*

One example - conflict between enablers

E.g., at a gas utility a staffing decision (HR), a policy, and a performance reward (punishment) collectively harmed the process

- HR – Outsourced Level 1 Customer Service Reps to BPO provider
- Policies and Rules – Level 1 CSRs must escalate certain cases (e.g., disconnection) to Level 2 CSRs employed at the utility
- Motivation and Measurement – Outsourcer is hit with a *financial penalty for every escalation!*
- *Outcome* – Discuss with your colleagues the likely outcome

Assessment by Enabler generates potential features for the To-Be

Workflow:

- Resource not available to Requestor until after *all* classification and tagging is complete, even though classification and tagging is unnecessary in many/most cases because the US Library of Congress and British Library do it and make it freely available to other libraries.

*To-Be **potential feature** (an idea) – make Resource available immediately, then do classification and tagging only if necessary, first checking if other libraries have done it*

IT:

- Three separate core systems lead to manual copying of data from system to system, often through "shadow systems."

To-Be potential feature – automated data replication

- Functional richness of core systems leads to overcomplexity

To-Be potential feature – identify the subset of features are really needed, and only use those

Assessment by Enabler generates ideas for the To-Be

Motivation & Measurement:

- Because work is so granular, no one is motivated by the performance of the whole, which is not even measured.

To-Be potential feature – develop relevant end-to-end metrics, and develop role and workgroup metrics to assess our impact on professional staff

Human Resources:

- Acquisition tasks don't require a skilled, higher cost Records Manager – Agency staff could do much more, RMs could do higher value work.

To-Be potential feature – Assign authority for higher-value work to Agency staff

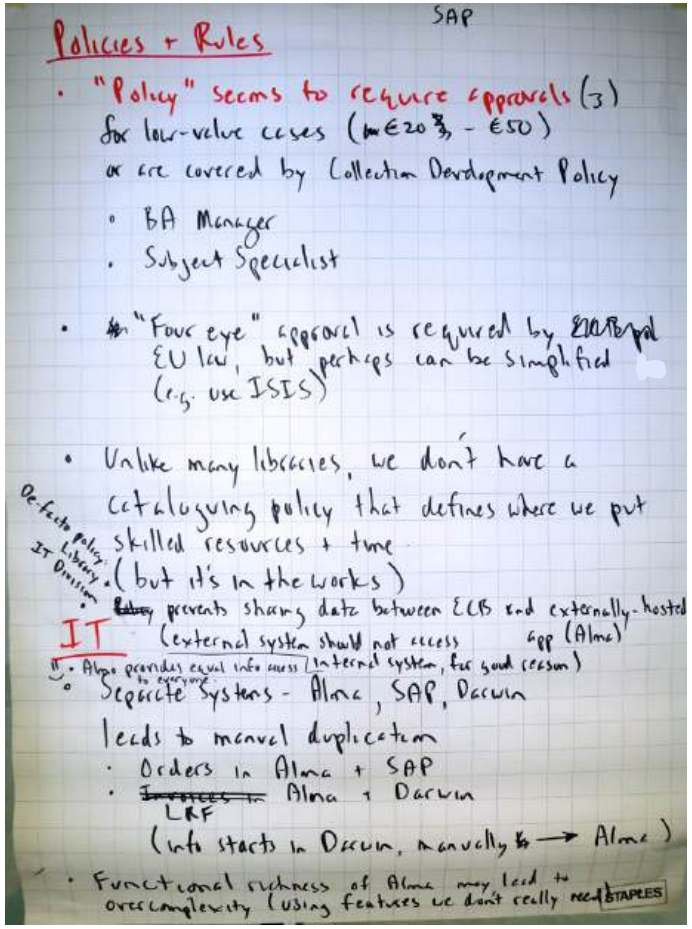
Policies & Rules:

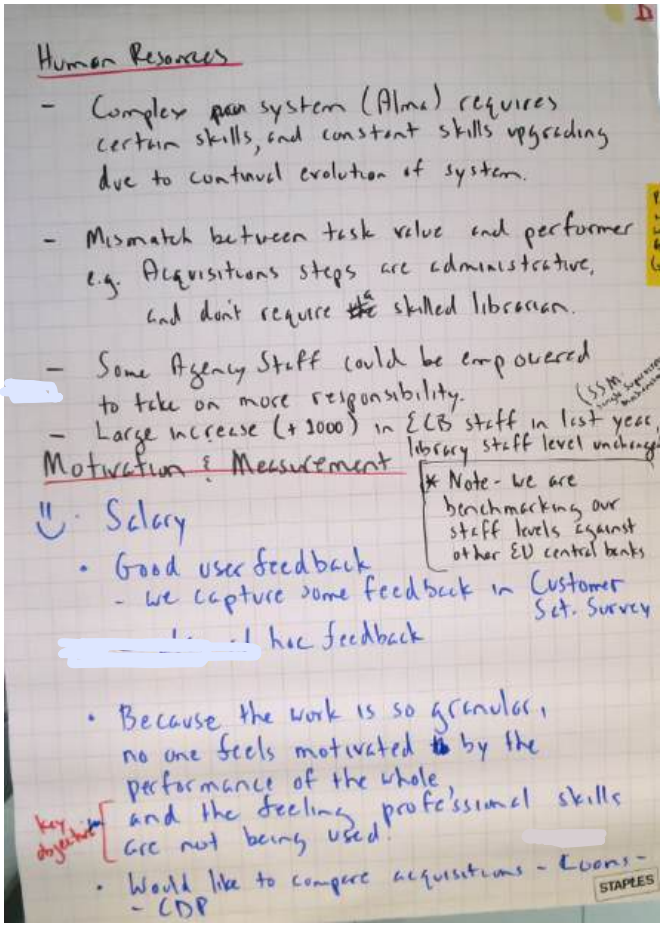
Three *(3!)* approvals required for low-value (€20 - €50) cases

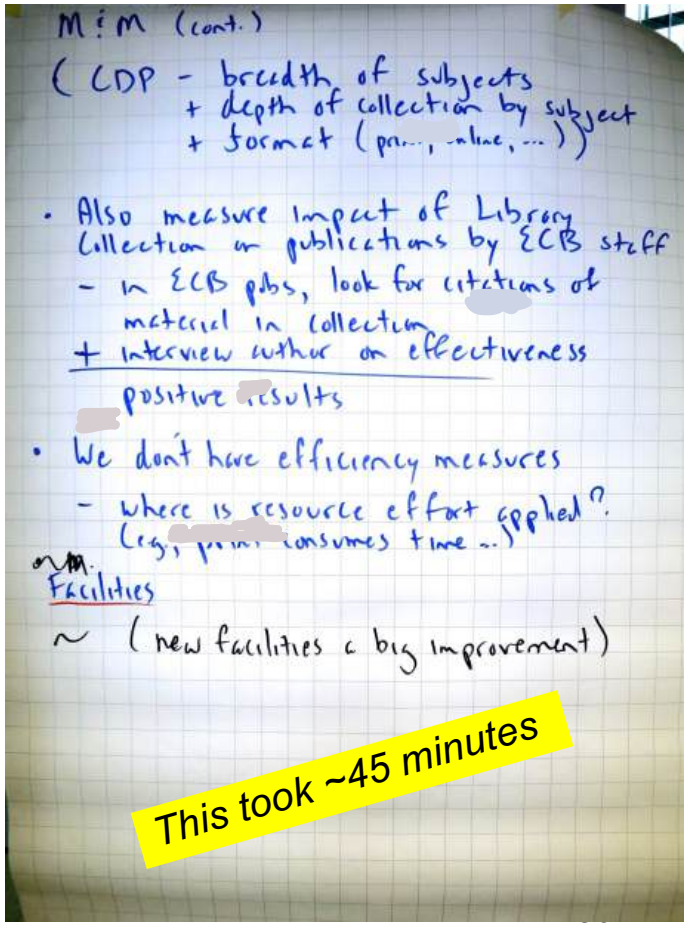
To-Be potential feature – revise policy to reduce approvals, eliminate them entirely for low-value cases

The value of a framework,

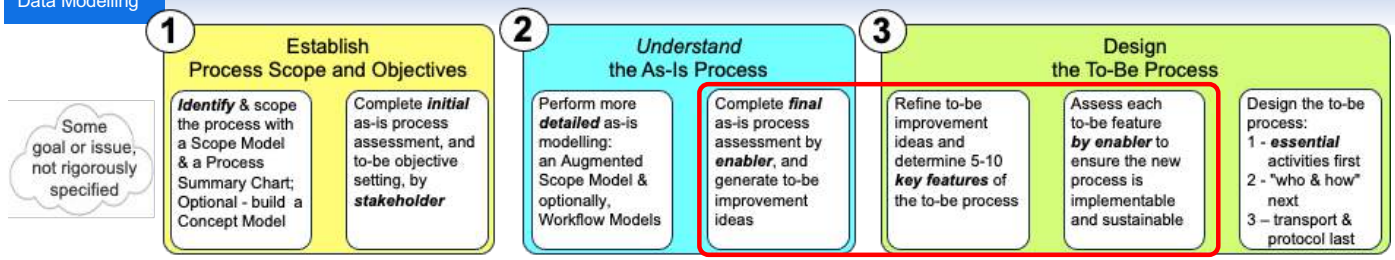
Give people a framework, go through it point-by-point,
and they will *quickly* identify factors that would have been *missed*.







Assess by enabler, establish 5-10 to-be features, assess each feature by enabler



A *feature* is a significant *change* or *improvement* to the process, or a significant *factor* in the design of an all-new process.

Enabler-based assessment of the *as-is process* generates ideas for the *to-be process*.

	Assessment:	Features:	
Motivation & Measurement	Sales Reps motivated entirely by commission, with no motivation to return and submit Service Orders	~~Increase Rep's commission for early submission~~	Rejected by execs.
		New Sales Assistant role to enter Service Orders	A *feature*.
Human Resources	Order Capture and Order Submission are not effective uses of a Sales Rep's time	Service Order entry directly by Customer	Another *feature*.
		New Sales Assistant role to enter Service Orders	Same *feature* again.

Then, assess each Feature – what changes are needed, enabler by enabler, to make this feature work?

Feature	Process Design	Info. Systems & Tech.	Motivation & Measurement	Human Resources	Policies & Rules	Facilities (or other)	*Feasibility & Notes*
Direct Service Order entry by Customers	Need to get the Service Order from the server to the Engineering Supervisor for assignment, and then to Engineer for assessment Customer review?	Obviously, all the Web stuff Integrated Service Order DB Workflow functionality? What format for Customer sketches?	Commission? What impact on commissions for current sales force?	Displacement of current Sales Reps? What are expectations for freed-up Sales Rep time? Customer training?	Will all Customers have access to this?	Electronic orders may free up space currently used for bins, boards, etc..	Highly feasible. What will Customer and Sales Rep reaction be?

Avoids unanticipated consequences!

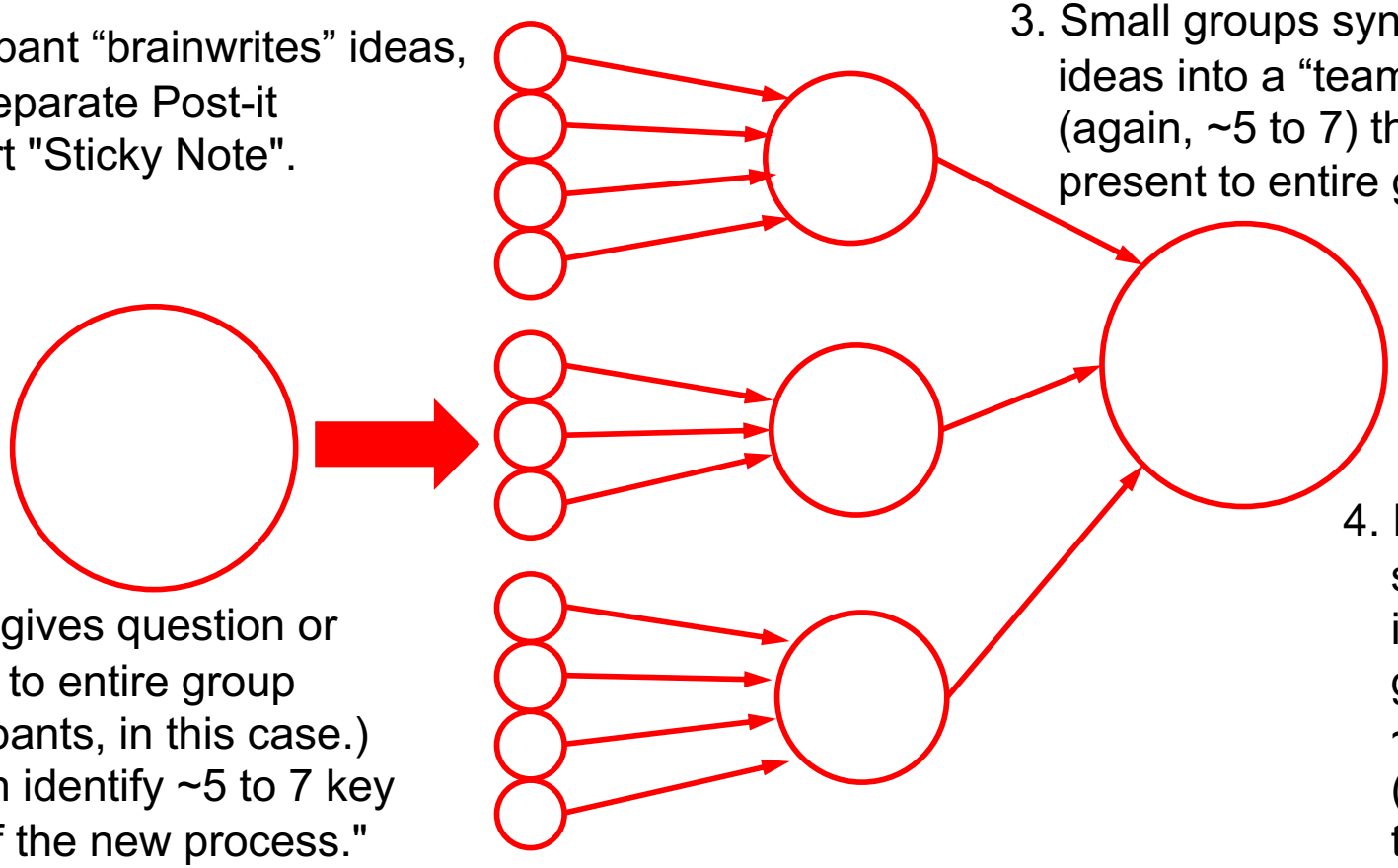
Determine to-be process key features – consensus approach

Use "brainwriting" and "big wheel, little wheel" facilitation

- Proven to generate *more* ideas / more *diverse* ideas
- Easier for *everyone* to contribute

2. Each participant “brainwrites” ideas, each on a separate Post-it or Lucidchart "Sticky Note".

3. Small groups synthesise ideas into a “team effort” (again, ~5 to 7) then present to entire group.

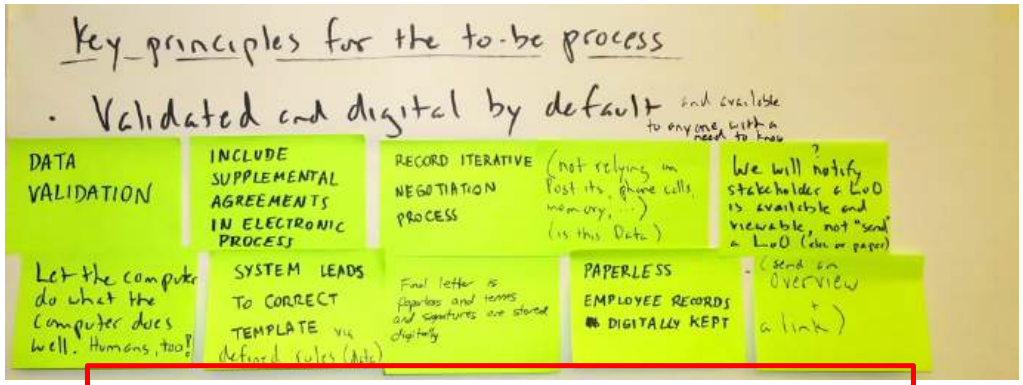


1. Facilitator gives question or instruction to entire group (11 participants, in this case.) "Let's each identify ~5 to 7 key features of the new process."

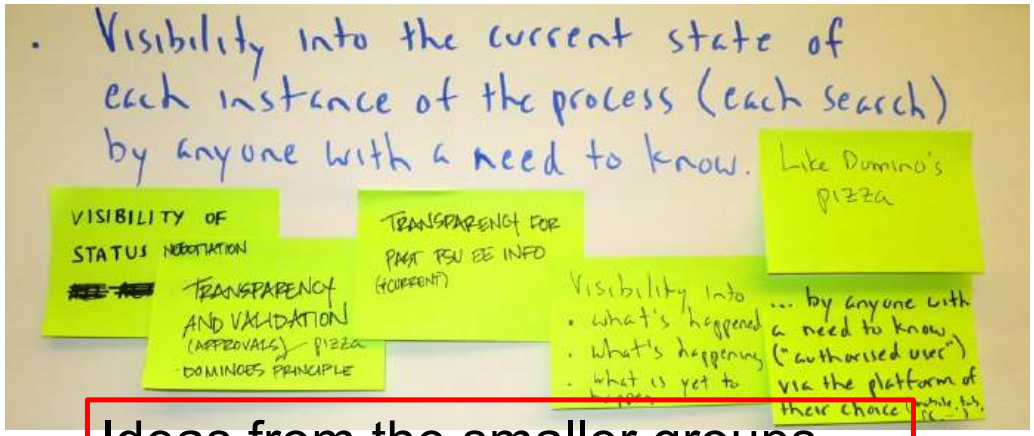
4. Entire group synthesises ideas into a group effort, ~5 to 7 *features* (rarely more than 10)

Example – determining features of the to-be process

Synthesis of features from group suggestions…



Ideas from the smaller groups…



Ideas from the smaller groups…

Five of seven features determined by the team

1. Data digital by default, validated and captured at source, and suitable for all downstream use.
2. Visibility into the current state of each instance of the process (each faculty search) by anyone with a need to know.
3. Separate the “need to approve” from the “need to be informed.”
4. Each search will follow a defined and visible workflow.
5. *The process will be designed for digital signatures **only – no fallback!***

Lucidchart / Lucidspark, Miro, or even Google Jamboard are perfect for a brainwriting session like this.

Features usually focus on one enabler, but involve all

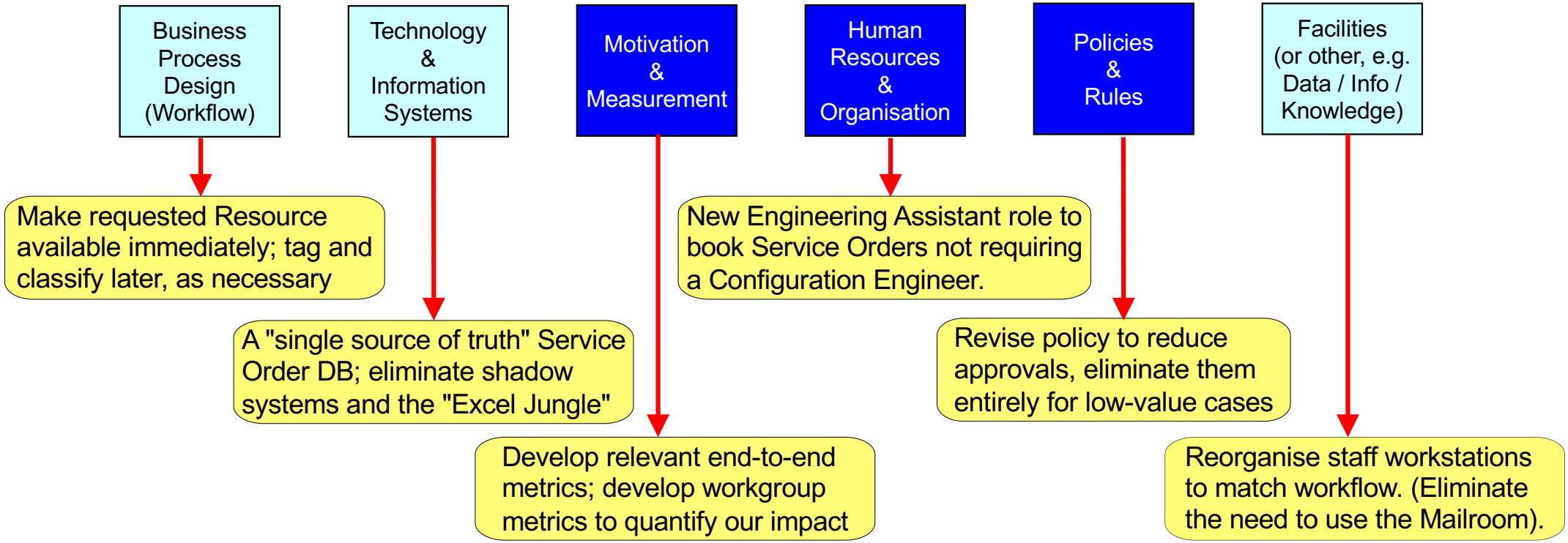
Reminder:

A *feature* is a significant *change* (*improvement)* or *factor* in the design of a new process.
Can be implemented one at a time.

Intent:

- "Don't sweat the small stuff" – focus on significant ideas.
- Avoid "Big Bang" implementations – implement feature-by-feature

Assessment by Enabler (and other techniques) generates ideas – some become *features*:



Another example – assessing each to-be feature, enabler by enabler

Intent:

- Ensure each feature is *implementable* and *sustainable*
- Avoid *unanticipated consequences* through a holistic assessment

For each *feature,* ask…
"What needs to change in *this specific enabler* to make this *feature* work?"
***Changes in multiple enablers are usually needed for each feature.

Feature	Process Design	Info. Systems & Tech.	Motivation & Measurement	Human Resources	Policies & Rules	Facilities (or other)	*Feasibility & Notes*
Assign authority for higher-value work to Support Staff rather than having it all done by Senior Records Managers.	Need to decide whether we can auto-route requests to the appropriate staff member, or if all should go to a Senior Records Manager for routing	Current systems are much too complex for most cases, especially the ones that would now go to Support Staff. Need to isolate and only display essential functions	We MUST adjust the performance measures of Support Staff to ensure they are not penalised for taking on additional responsibility	Revise job descriptions for Support Staff as necessary. Provide additional training in Records Management functions and the RM System	Current policies dictate that all categorization and classification work be carried out by Records Managers – this will have to change. Some regulations may be a factor	Some Support Staff will be moved closer to Records Managers, but this is a minor change	Highly feasible if we can resolve Policy issues. Support Staff are very positive about the opportunity, and Records Managers look forward to more time for high-value work.

This feature required change in *all six* enablers, especially M&M and P&R!

A richer example – first, describe the feature (page 1 of 2)

A surprise benefit – invaluable during training and roll-out.

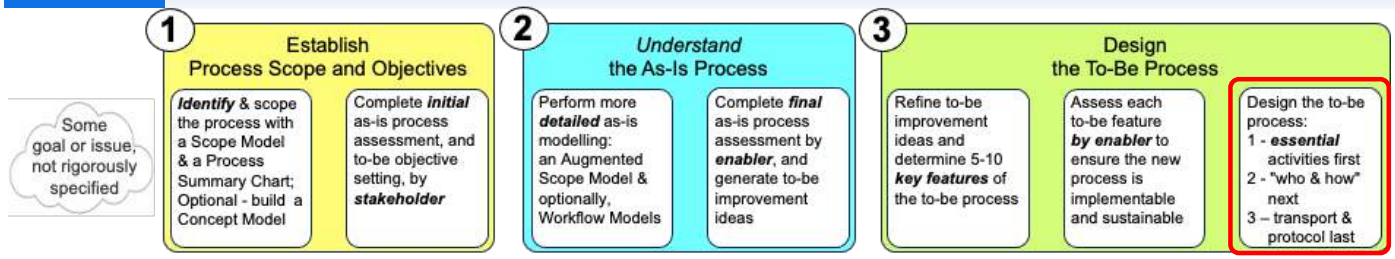
Feature name (A feature is a particular characteristic or improvement in the to-be process)
Forensic strategy (“applying science at the front end”)
Description
A Senior Scientist, typically the Case Manager, will meet with the Submitting Officer and develop a case strategy specifying which avenues of investigation, and which items and tests are most likely to yield the needed results in the least time with the least effort. The goal is to do this for as high a percentage of cases as possible. This is the first decision point in another characteristic, *multiple decision points.* Visually, this is the first stage in a funnel, in which the work being performed on a case is continually reduced as new facts arise.
Issues addressed
There is a tendency for the Customer (the police) to submit all possible items, and request all possible tests, or at least submit more items for more tests than are necessary or justified. This is known as “forensicating” a case and is ironically a primary cause of the delay and expense that the customer is unhappy with. Currently, Forensics accepts all items and performs all requested tests through to completion. In some cases, the suspect has become the accused and then the defendant, and has been convicted and incarcerated, yet testing continues.
Anticipated outcomes / benefits
For the Customer – deliver a positive result in less time, at less cost. For Forensics – free up resources by reducing submissions, and performing fewer tests on fewer items, thereby providing better throughput for all cases. In the future, Forensics will only perform those tests that will help, and which will stand up in court because we can say “we chose these tests for these reasons.” On an ongoing basis the customer will become more aware of the avenues that are most effective.

Then identify requirements to implement each feature (page 2 of 2)

Eight features assessed in a single five-hour session!

Enablers	
Process Design	Performers ("actors"), tasks, sequence, dependency • Senior scientist "meets with" appropriate scientist, not necessarily in person • Assessment and agreement and recording of *requirement* which is not contracted yet. • The requirement must be made available to the Process Manager, who will assess it with respect to current capacity. • The Case Manager and Process Manager will then negotiate and refine the requirement. They will then agree on "what and when" and commit capacity, which might involve another provider.
Information Systems & Technology	Systems, automated support, data and Information, comm. • Capture requirement • Real-time view into work-in-progress and committed capacity (Forensics' and subcontractors)
Motivation and Measurement	Measurement, assessment, consequences • The Process Manager will be measured on accurately estimating capacity and throughput. • The Process Manager makes a commitment for Forensics, and will be measured on having done the least to get the necessary result. ("lean consumption")
Human Resources	Recruitment, placement, education, roles, matching task to role • New front-end role for scientists • Process Manager role • Provide service 24x7 will impact some staff. • Recruitment, recognition, and reward are fundamental to making this work
Policies and Rules	Internal: policies & guidelines. External: laws and regulations • The overall submissions policy must be revised to reflect forensic strategy vs. "take it all." • Investigate legal consequences of forensic strategy. • Mechanism to protect the individual scientist from pressure. ("Forensics, not the individual scientist" – this is a corporate decision, not a personal decision) • Scientists can't make commitment without the Process Manager. • A 10 minute phone call and a 4 hour conference both constitute delivery of a service. A request to confer with a Case Manager constitutes contract initiation.
Facilities and Equipment	Physical accommodations, layout, equipment, furnishings • Some place to meet – in person, teleconference, …

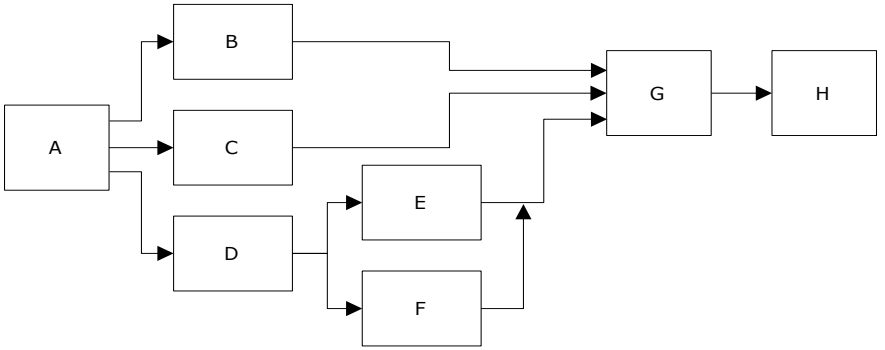
Design to-be process – overview

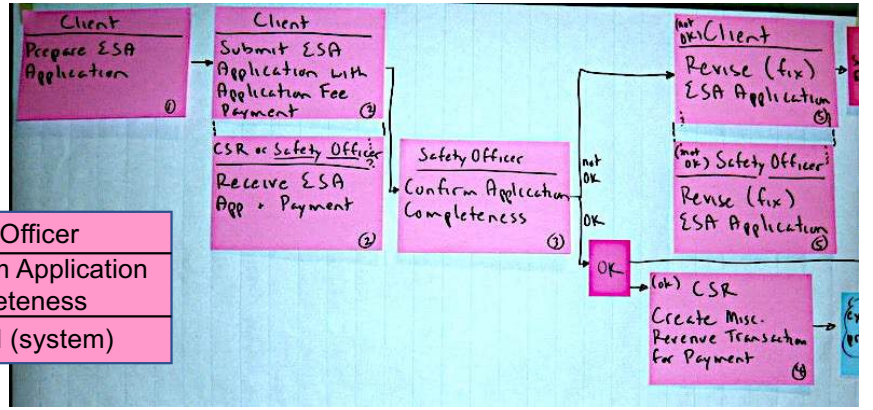


Key points:

- As with the as-is process – *"What first, who and how later"*
- Design around *essential* steps, not *administrative steps*

- Use an Augmented Scope Model to determine *what* the essential activities are
- Next, factor in *who* will perform each activity, then *how*
  - a person as a manual activity
  - a person interacting with a system, e.g. a use case
  - a system, e.g., RPA (Robotic Process Automation)
- Link essential activities by dependency – a PERT chart
- Adjust – e.g., verify activity is assigned to the correct role
- Only then redraw as a swimlane diagram
- *Finally,* add non-value-added but necessary activities:
  - transport, record keeping, notification, etc.
  - ensure any approval steps are *really* necessary ("Don't confuse notification with approval.")





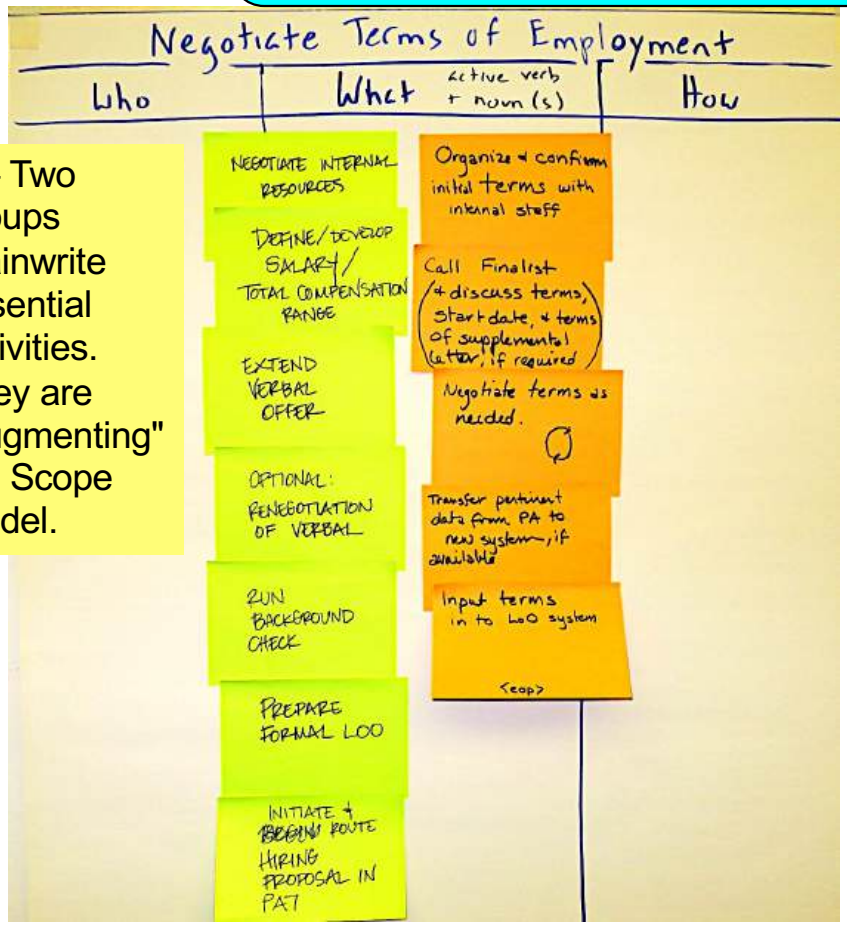
Who:	Safety Officer
What:	Confirm Application Completeness
How:	S-MAN (system)

4 – Design to-be process – the details – identify essential activities

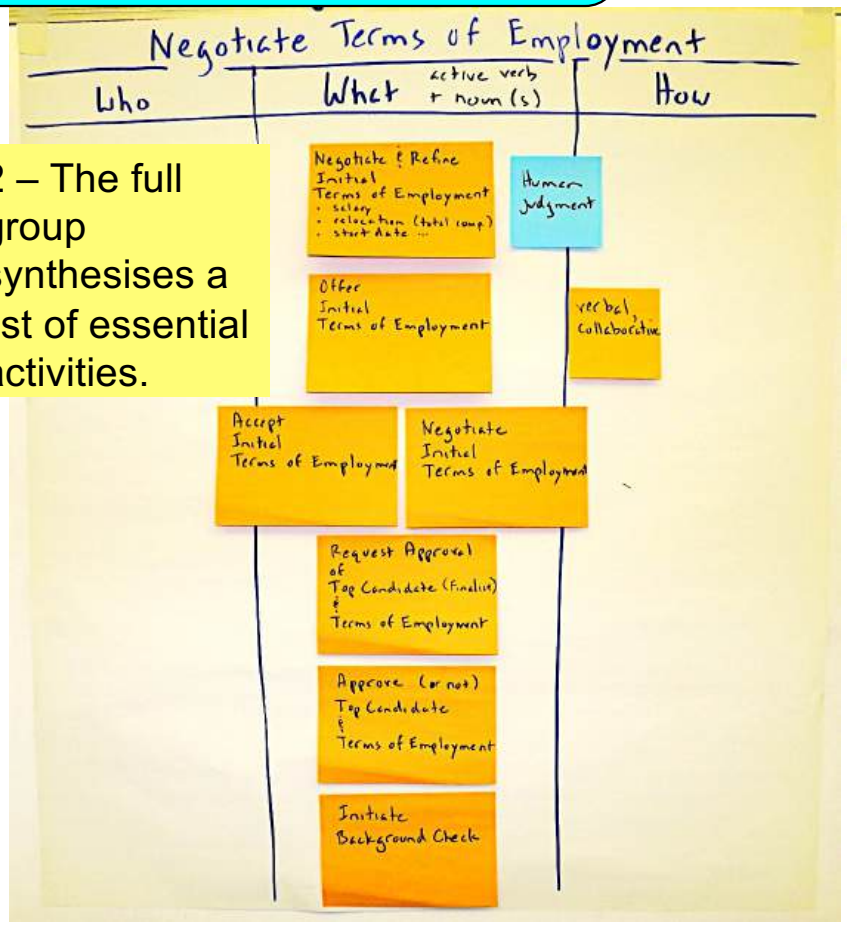
Recruit, Hire, and Onboard Employee

Prepare to Recruit	Recruit Applicants	Evaluate Applicants & Select Finalist	Negotiate Terms of Employment	Finalise Terms of Employment	Onboard Employee

1 – Two groups brainwrite essential activities. They are "augmenting" the Scope Model.



2 – The full group synthesises a list of essential activities.

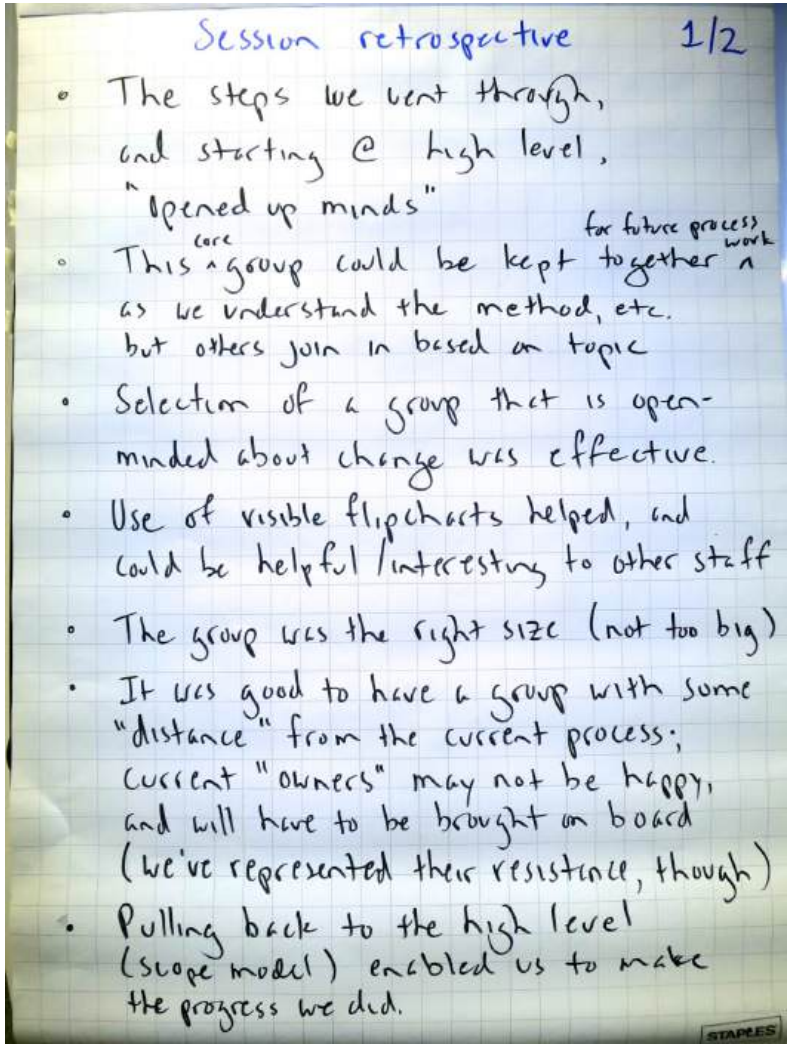


Lucidchart version

For each essential Activity, add "Who," "How," and lots of "Notes"

- We have the core of the to-be process design
- Going immediately to a Swimlane Diagram would be *overwhelming!*
- But now, developing the to-be flow model (swimlane diagram) is straightforward – *We Can Do It!*
  We have:
  - actors (swimlanes)
  - steps
  - how the steps will be done
  - sequence
    (approximate, but OK for now)

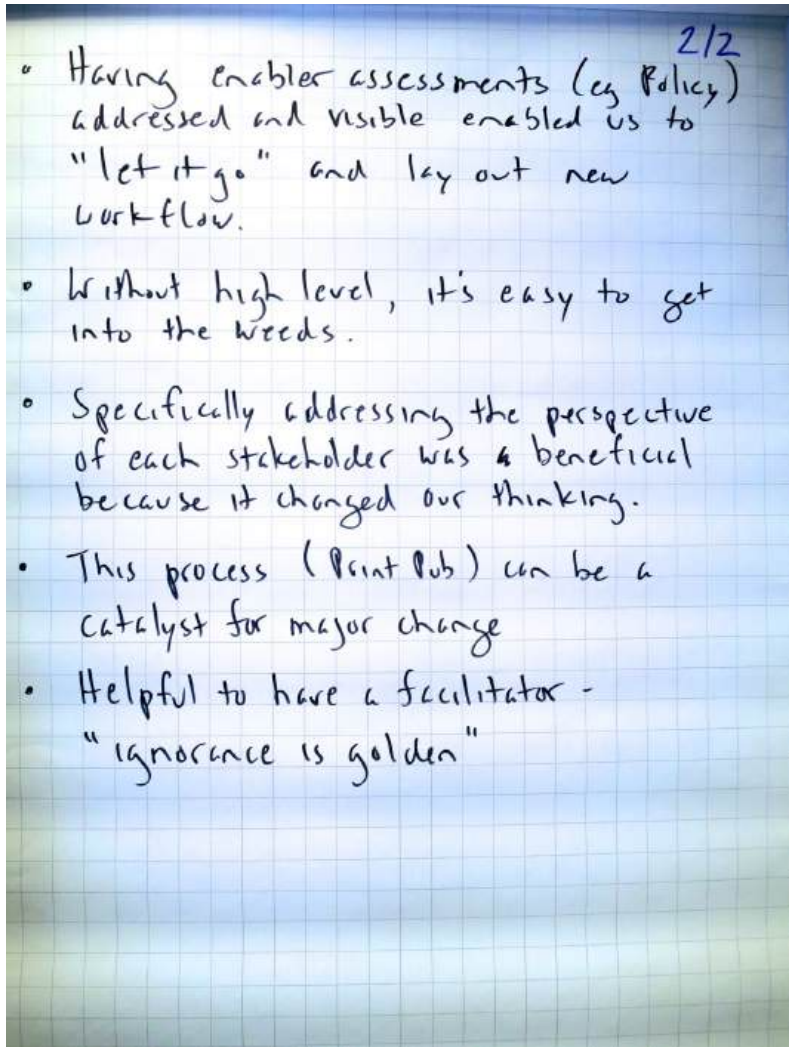
Final observations from session retrospective, 12 people



Session retrospective 1/2

- The steps we went through, and starting @ high level, "opened up minds"
- This core group could be kept together for future process work as we understand the method, etc. but others join in based on topic
- Selection of a group that is open-minded about change was effective.
- Use of visible flipcharts helped, and could be helpful/interesting to other staff
- The group was the right size (not too big)
- It was good to have a group with some "distance" from the current process; current "owners" may not be happy, and will have to be brought on board (we've represented their resistance, though)
- Pulling back to the high level (scope model) enabled us to make the progress we did.

STAPLES



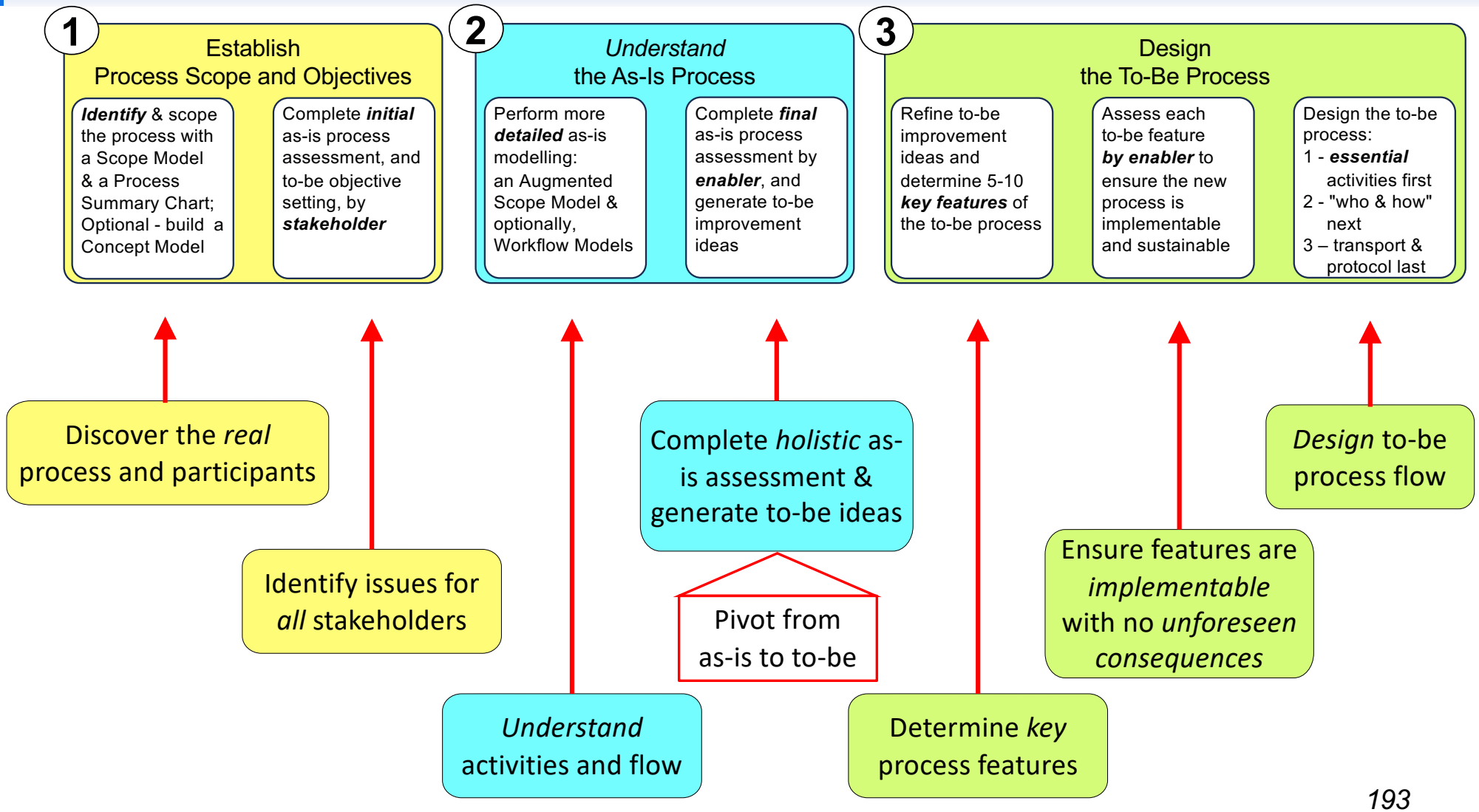
2/2

- Having enabler assessments (eg Policy) addressed and visible enabled us to "let it go" and lay out new workflow.
- Without high level, it's easy to get into the weeds.
- Specifically addressing the perspective of each stakeholder was a beneficial because it changed our thinking.
- This process (Print Pub) can be a catalyst for major change
- Helpful to have a facilitator - "ignorance is golden"

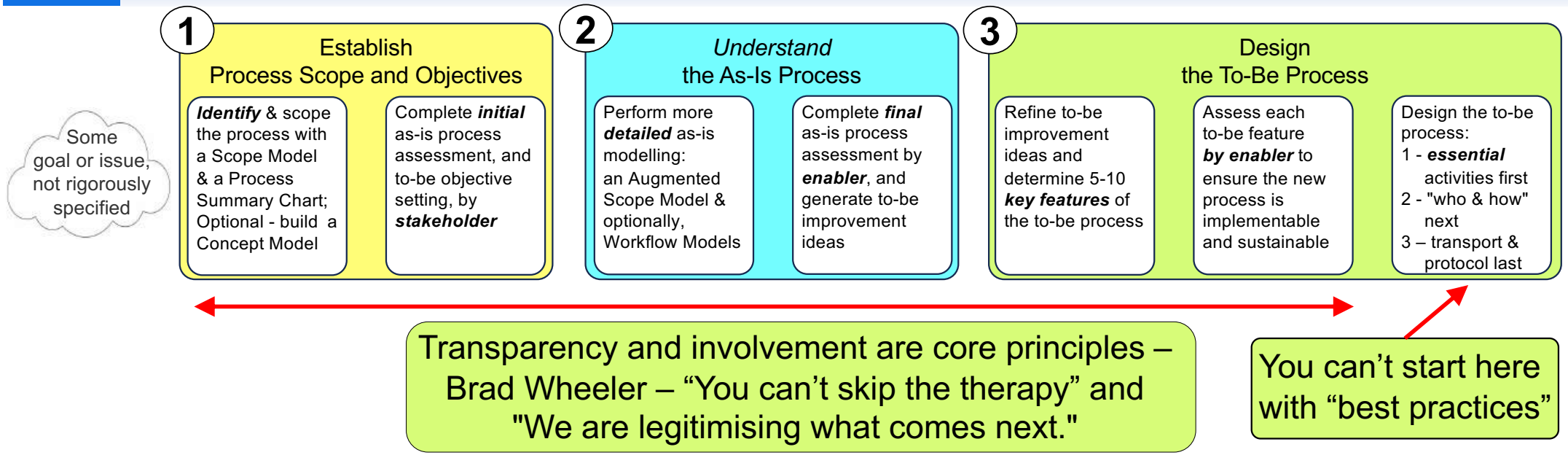
Final thoughts from session retrospective

- The steps we went through and starting at the high level "opened up minds."
- Use of visible flipcharts helped, and could be helpful / interesting to other staff.
- Pulling back to the high level (Scope Model) enabled us to make the progress we did.
- Having enabler assessments (e.g., Policy) addressed and visible enabled us to "let it go" and lay out new workflow.
- Without the high level, it's easy to get into the weeds.
- Specifically addressing the perspective of each stakeholder was beneficial because it changed our thinking.
- Helpful to have a facilitator – "ignorance is golden."

Every phase contributes to the goal – don't skip any!



Remember – "It's a process!"



Making the new process sustainable:

- Alignment of *all* enablers, especially *Motivation & Measurement*, *Human Resources & Organisation*, and *Policies & Rules*
- Visibility of the process – the *whole* process, right down to *job aids*
- *Training* in the new process for current and new staff
- Time for *each feature* of the new process to *take hold* before more change – *continuous* change should mean *regular* but not *constant* change

Process - Data Synergies

Course Topics

1. Requirements Definition
   - Goals, Issues, and the Return of Modelling
   - Case Study - Integrating the Techniques
2. Business Process Fundamentals
   - Five Things You Need to Know
   - Discovering, Scoping, & Assessing Your Processes
3. Concept Modelling Fundamentals
   - E, R, A - A Concept Model's Essential Components
   - Drawing Your Model for Maximum Understanding
4. Business Process Workflow Modelling & Design
   - Five Core Guidelines for Great Swimlane Diagrams
   - Facilitating a Process Mapping Session
   - Assessment of the As-Is and Transition to the To-Be
5. The Process-Data Connection
   - The Natural Synergy between Process & Data Models
   - Process-Data Synergies in Modelling, Analysis, & SW

"Process people" make "process" far too difficult

1 – No clarity on what "Business Process" means…

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*)

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

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

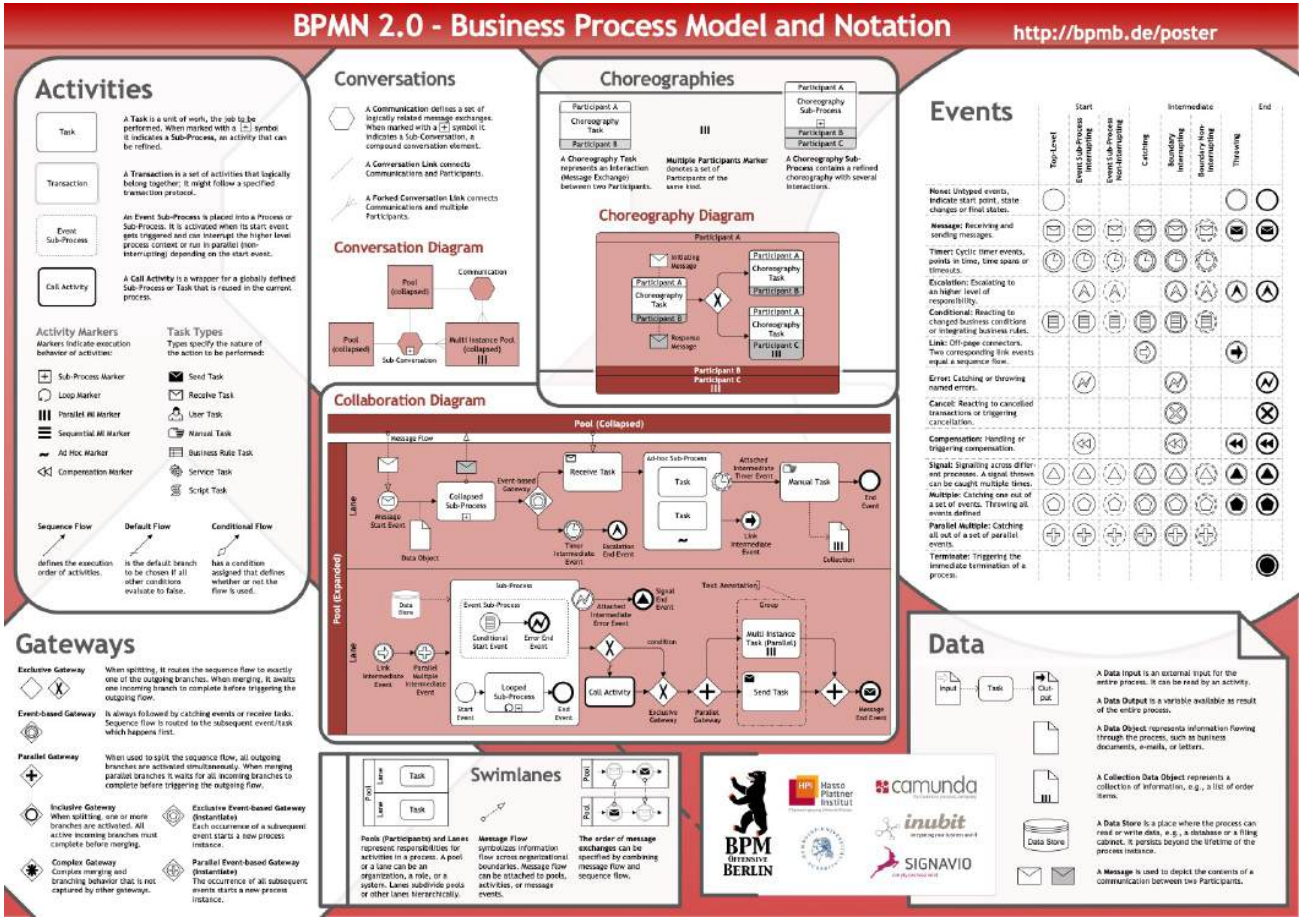
Seek balance – a "business process" lies between the extremes

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

The key issues – granularity and orientation

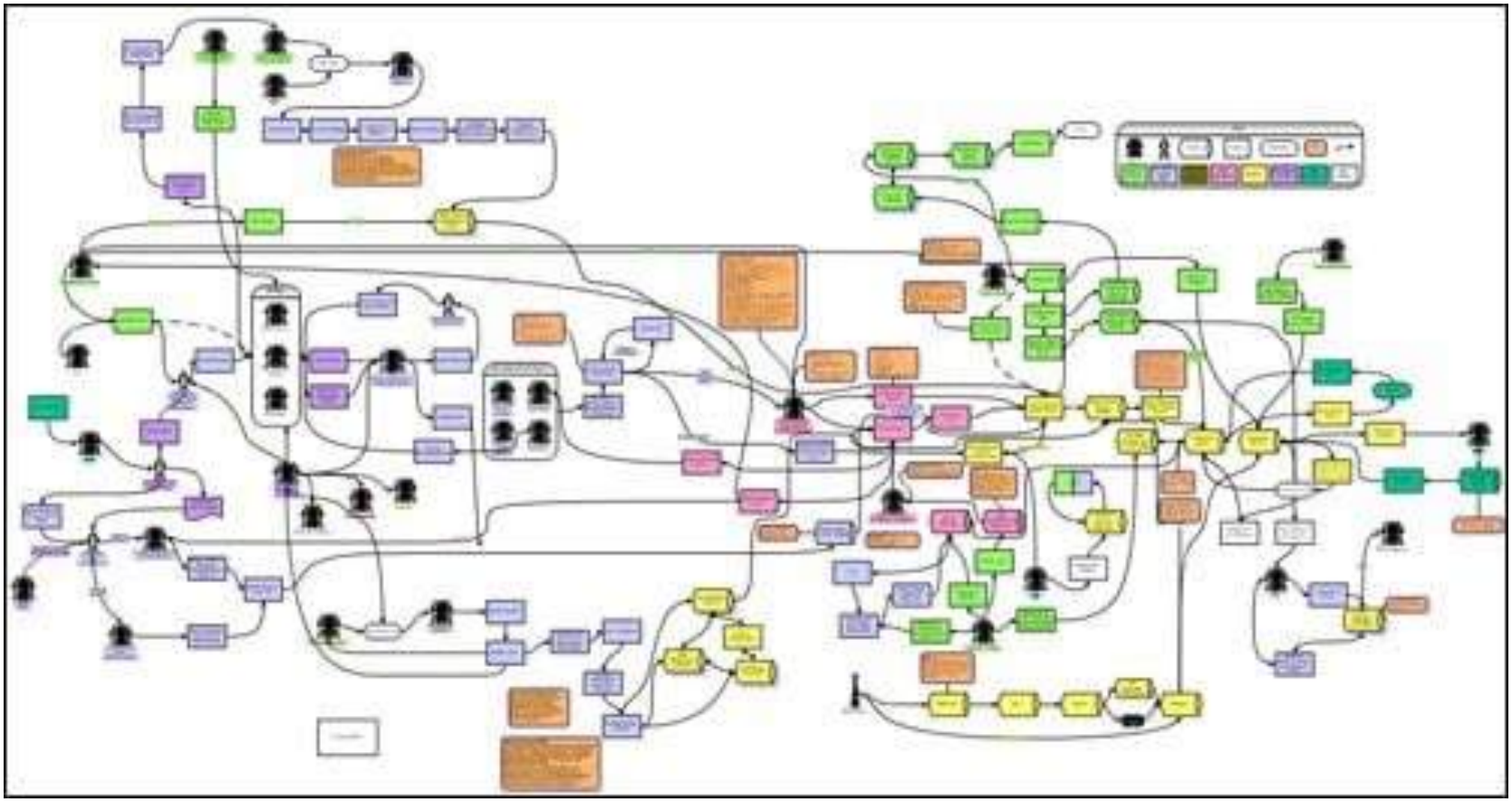
"Process people" make "process" far too difficult

2 – Technically oriented standards…



"Process people" make "process" far too difficult

3 – The sudden deep dive into detail…



"Data people" make "data" far too difficult

1 – Confusion between data modelling and database design…

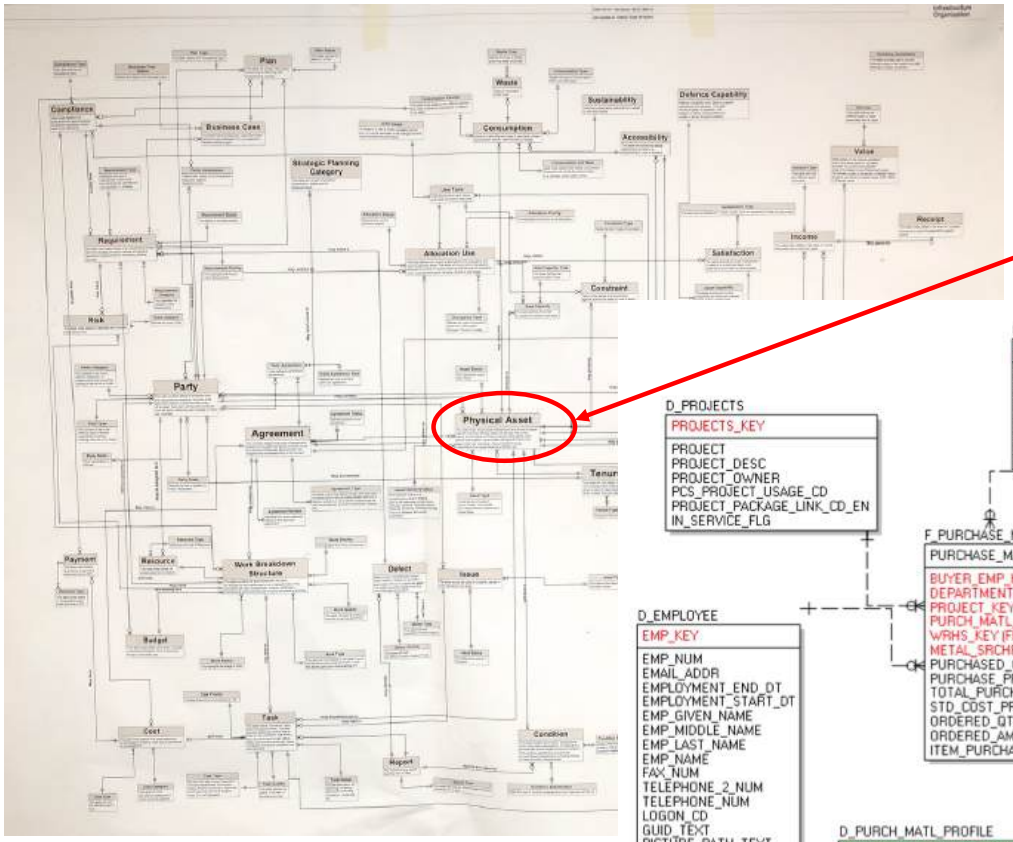
"Help – everyone hates our data model."

"Data people" make "data" far too difficult

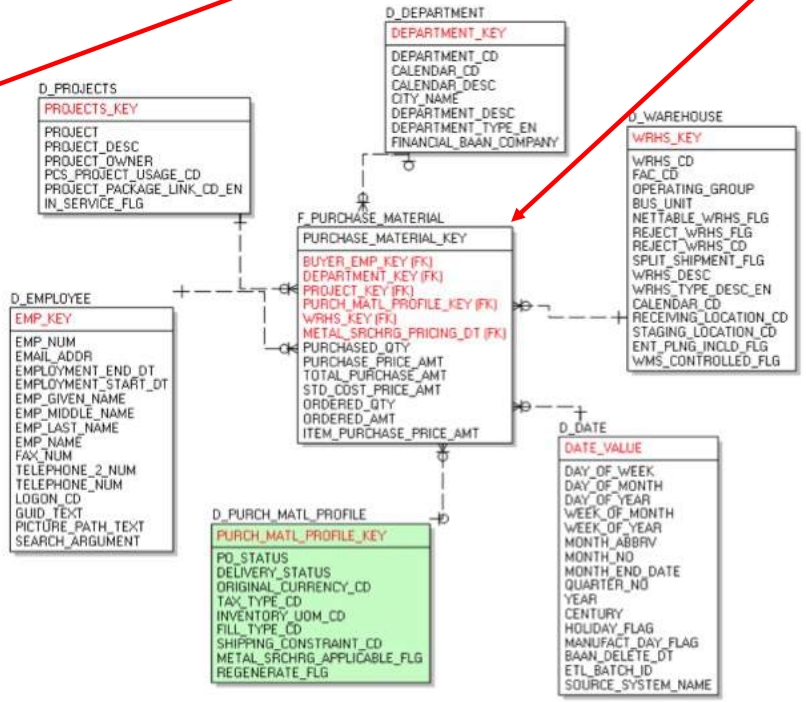
2 – Terrible diagramming…
A common error – *"the most important entity should go in the centre of the diagram."*

An excellent model *structurally*, but very difficult to follow – *no sense of direction.*

Concept Models / ER Models should be drawn top-down by dependency.



"Fact" in the middle - fine for Dimensional, terrible for E/R



Specifics – contextual, conceptual, logical data model

For 18 years, my most plagiarised diagram!

1 Contextual (Scope)	2 Conceptual (Overview)	3 Logical (Detail)
Agree context or "big picture" – the scope in terms of topics or subjects that are in or out, plus core terms and definitions • May be a simple block diagram of topics/subjects, or primarily textual (a list) • Optional – not necessary on smaller projects	Agreement on basic concepts and rules • Ensures everyone is using the same vocabulary and concepts before diving into detail • Overview: main entities, attributes, relationships, rules • Lots of M:M relationships • Relationships show cardinality • No keys • Few or no reference entities • Unnormalised – most M:M relationships unresolved, many attributes will be multi-valued, redundant, and non-atomic • Verified directly by clients plus other techniques: Use Cases… • A "one-pager" • 20% of the modelling effort	Full detail for physical design • Provides all detail for initial physical database design and requirements specification • Detailed: ~ 5 times as many entities as the conceptual model • M:M relationships resolved • Relationship optionality added • Primary, foreign, alternate keys • Lots of reference entities • Fully normalised – no multi-valued, redundant, or non-atomic attributes. All attributes defined and "propertised" • Verified by other means: sample data, report mockups, scenarios, … • May be partitioned • 80% of the modelling effort

3 – No clarity on different types of models for different perspectives

The Lost Art of Conceptual Modeling

Alec Sharp, Acetta LLC
alec.sharp@acetta.com or
asharp@clariteq.com

I've been making this point for over 20 years…

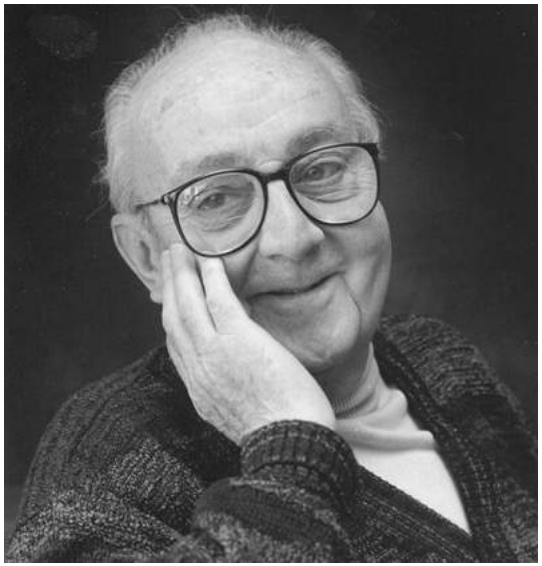
- 2004 DAMA – The Human Side of Data Modeling
- 2005 DAMA Symposium panel
- 2006 DAMA – Lost Art of Conceptual Modeling

CLARITEQ acetta

So, of course, they usually don't understand each other

A core idea – "essential" models

"All models are wrong, but some are useful."



George E. P. Box
1919–2013

Two especially useful models

- Business Process Scope Model
- Business Concept Model
  (a.k.a Conceptual Data Model)

Both are "essential" – they show the essence – the "what" – of a subject with no reference to who, how, why, etc.

Essence and Accident?

Essential:
1) ~~absolutely necessary; extremely important.~~
2) something's basic or most important characteristics;
the intrinsic, inherent, or fundamental nature of something

Cup:
The *essential* characteristics:
a round, handheld container
for drinking from.
What it is.

The *accidental* characteristics:
ceramic vs. bamboo, handle or not, …
How it is designed or made.

No Silver Bullet

Essence and Accidents of Software Engineering

Frederick P. Brooks, Jr.
University of North Carolina at Chapel Hill

Fashioning complex conceptual constructs is the *essence;* *accidental* tasks arise in representing the constructs in language. Past progress has so reduced the accidental tasks that future progress now depends upon addressing the essence.

Of all the monsters that fill the nightmares of our folklore, none terrify more than werewolves, because they transform unexpectedly from the familiar into horrors. For these, one seeks bullets of silver that can magically lay them to rest.

The familiar software project, at least as seen by the nontechnical manager, has something of this character; it is usually innocent and straightforward, but is capable of becoming a monster of missed schedules, blown budgets, and flawed products. So we hear desperate cries for a silver bullet—something to make software costs drop as rapidly as computer hardware costs do.

But, as we look to the horizon of a decade hence, we see no silver bullet. There is no single development, in either technology or in management technique, that by itself promises even one order-of-magnitude improvement in productivity, in reliability, in simplicity. In this article, I shall try to show why, by examining both the nature of the software problem and the properties of the bullets proposed.

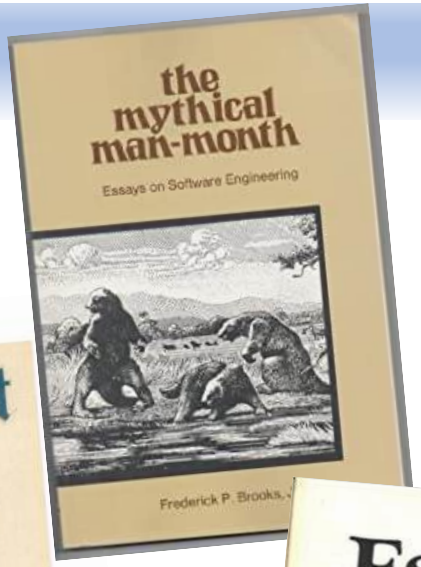
Skepticism is not pessimism, however. Although we see no startling breakthroughs—and indeed, I believe such to be inconsistent with the nature of software—many encouraging innovations are under way. A disciplined, consistent effort to develop, propagate, and exploit these innovations should indeed yield an order-of-magnitude improvement. There is no royal road, but there is a road.

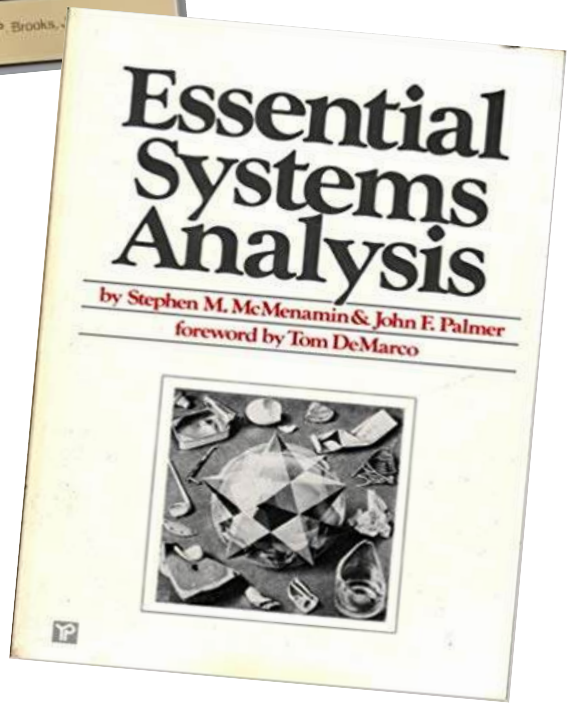
The first step toward the management of disease was replacement of demon theories and humors theories by the germ theory. That very step, the beginning of hope, in itself dashed all hopes of magical solutions. It told workers that progress would be made stepwise, at great effort, and that a persistent, unremitting care would have to be paid to a disciplined cleanliness. So it is with software engineering today.

Does it have to be hard?—Essential difficulties

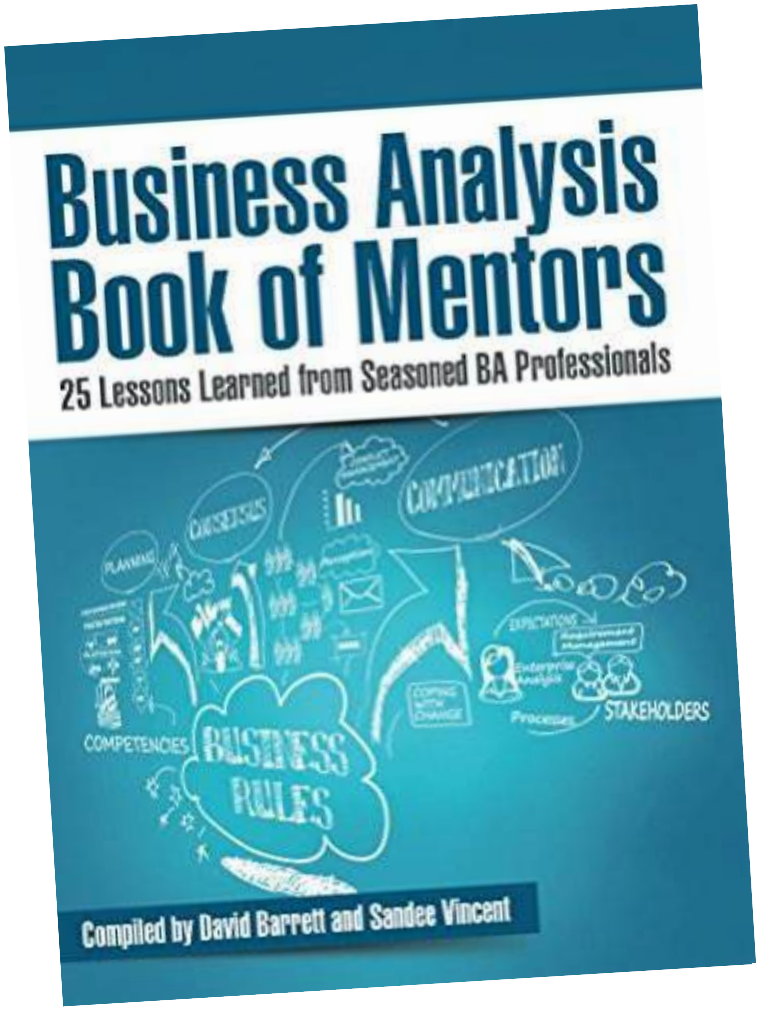
Not only are there no silver bullets now in view, the very nature of software makes it unlikely that there will be any—no inventions that will do for software productivity, reliability, and simplicity what electronics, transistors, and large-scale integration did for computer hardware.

COMPUTER





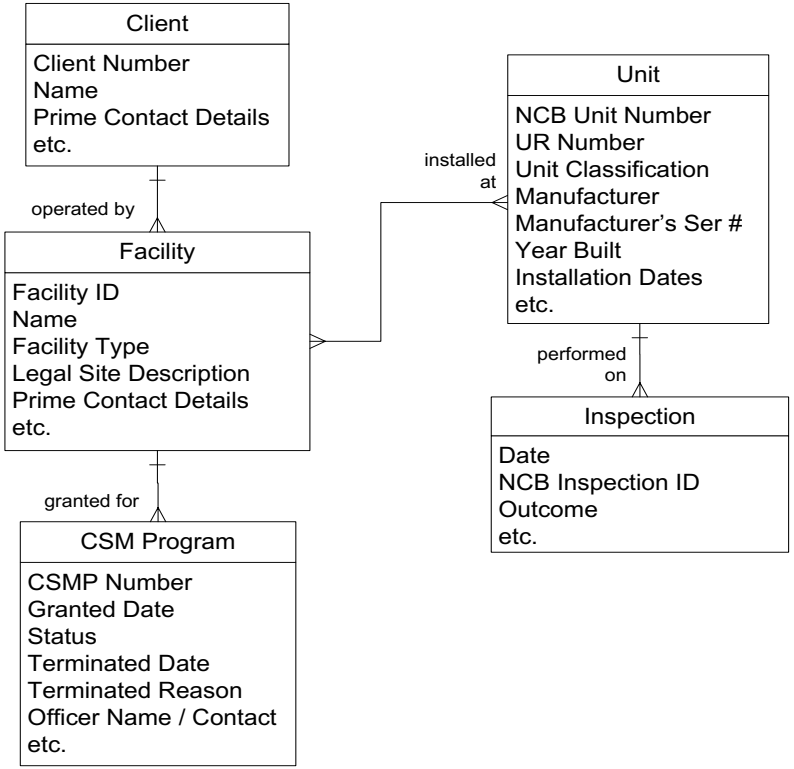
My chapter in the "BA Book of Mentors"



The premise of the book:

- 25 experienced BAs from around the world would each write a chapter on "The Most Important Lesson I Learned in my BA Career.'
- I knew mine instantly – separate the "what" from the "who, how, and why"
- In other words, separate the "essence" from the "accident"
- The "what" with no reference to "who" (role or organisation) or "how" (implementation or technology)
- Getting out of the who and how helps people find common ground

Concept Model – an Essential model*

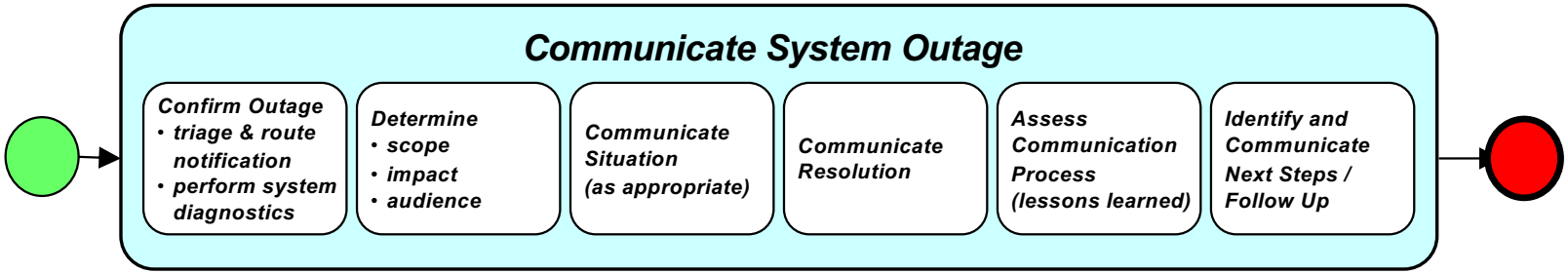


Concept Model:
a description of a business in terms of

- what *things* it needs to know about to operate – entities, business objects, classes, *things,* …
- what *facts* it needs to know about those things – relationships & attributes
- what *policies & rules* govern those things– definitions, constraints, and assertions

A *shared language* of the *nouns*
that are central to the enterprise.
Always start here!

Process Scope Model – an Essential* model



Triggering Event:

Notification of degradation or lack of Service
- internal system
- external provider
- calls to Service Desk

Cases:
- new
- recurring

Other factors:
- severity
- key operations periods / areas (registration, summer, course evaluation season)
- time of year
- time of day

Results:

Communications about the Outage and the progress on resolving it are delivered:
- internally and externally
- informally and formally

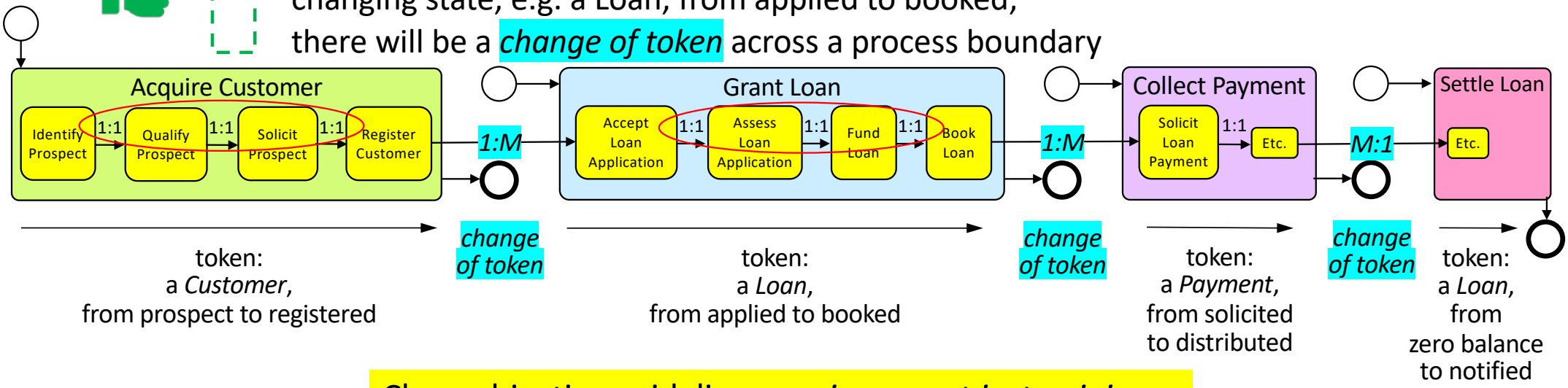
Final Results:

Service is restored and root cause is known (or is determined to be unknowable) and resolution is communicated:
- Externally ("good news")
- Internally ("cause & resolution)

Process Scope Model using "TRAC" -
what is the Trigger,
what are the Results,
what are the main Activities,
(7 ± 2 milestones, phases, or subprocesses,)
and *what* are the main Cases?

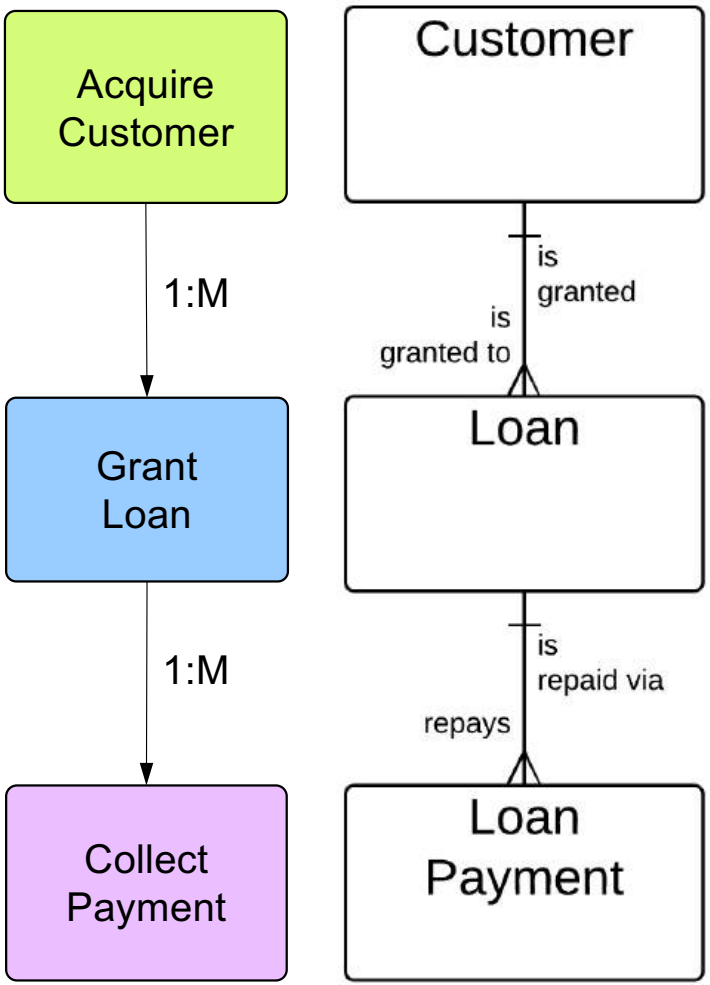
Reminder – nouns (entities) help identify processes

1. "Active verb – noun" naming that indicates primary result
2. Triggered by an event (decision, time, data) outside process' control
3. At the end are results that makes one or more stakeholders happy
4. In between are ~5 to 7 phases, milestones, or major activities
5. Activities linked *1:1* are probably part of the same process; a *1:M* or *M:1* connection between activities is probably a boundary
6. The same *token* moves through the whole process, changing state, e.g. a Loan, from applied to booked; there will be a *change of token* across a process boundary



Clear, objective guidelines – *science, not just opinion*

Correspondence to the Concept Model

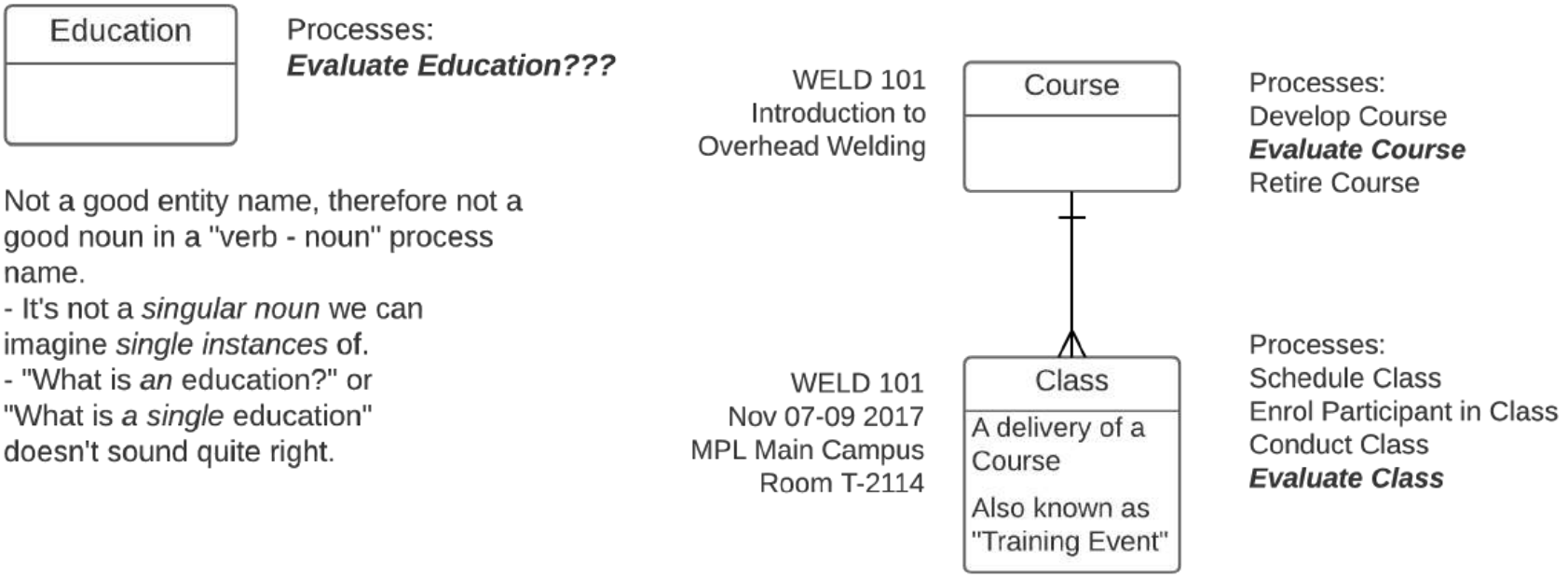


- The nouns in your verb-noun *process* name are most often the *entities* in your Concept Model, and each will usually have one primary *process*
- The relative number of process instances (e.g., 1:M or M:1) aligns with relationship cardinality
- This *does not* mean there is only one process per entity
  - Assess *Customer* Performance
  - Retire *Customer*
  - Merge *Loans*
  - Write Off *Loan*
  - …

Example – simple Concept Modelling to clarify the process

Analyst struggles to model “Evaluate Education” – timing disconnects, 1:M and M:1 connections within the process, token changes, …

A few minutes of Concept Modelling showed two distinct tokens and processes. “Education” was a “mushy noun.”



The key was asking "What do you mean by an *Education?*”

Never be afraid to ask "dumb" questions…

- Myth -

"I've got to have all the answers. I can't show my ignorance."

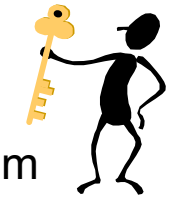
Just one more question, ma'am. Nothing too important...

There's one thing I'm not clear on...

Could we go over this just once more to be sure I've got it right?

- Reality -

You're paid to ***ask,*** *not to* ***know.***
Someone will be glad you did.
The number of different answers will surprise ***everyone.***
Classic example – "Case" in a justice system

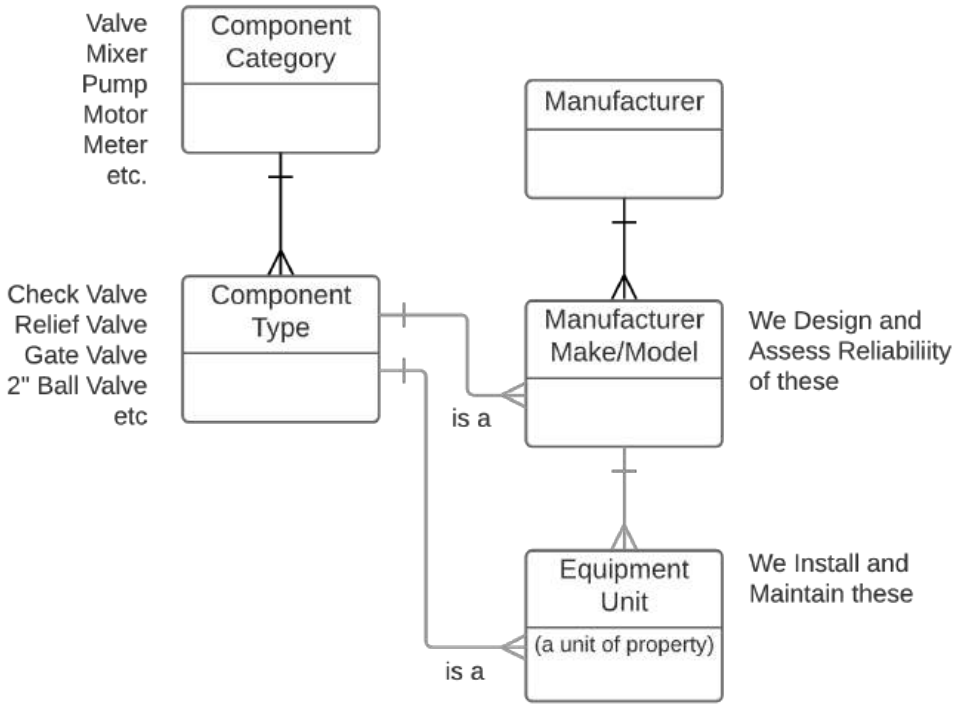


Lieutenant Columbo takes up Data Modelling

Similar example – Concept Modelling to clarify the process

A session to model the “Design Component” process at a pipeline operator is going in circles. Concept Modelling clarifies the company doesn't actually “design components,” they:

- Develop Component Type Specifications
- Approve Manufacturer Make/Model (“AML”)



An advanced example, if we have time – Is a new process concept viable?

Classroom tech support at major US research university

- Goal: “Uber-style” tech support for classrooms – when an Incident is raised in a Classroom, dispatch it to one or more appropriate Techs (qualified, available, assigned to the appropriate Support Unit) who will bid on it.
- Approximately 20 “assertions” described the planned state:
  - Each Tech may be badged for one or more Service Category Levels, and for each Service Category Level there may be one or more Badged Techs.
  - Each Tech may be assigned to one or more Support Units during a given time period, and for each Support Unit there may be one or more assigned Techs.
    A Tech can only be assigned to one Support Unit at a time.
  - An Incident for a particular Classroom can be raised by either a Customer (the “reporter” – Faculty, Staff, Tech, …?) or an automated Alert raised by an Equipment Unit located on a particular GP Classroom.
  - many more…
- The assertions led to the development of an ERD.
  Note – the complete “Concept Model”
  is the combination of the definitions, the assertions, and the graphic (ERD)

Example 5 – Assertions. Lots of assertions.

Classroom Support

Assertions, for review and validation:

- Support is provided by different Support Units (organizations) for different Service Levels (tiers) and different Service Categories (Computers, Audio-Visual, Learning Technologies, Networking, Scheduling, and Facilities.) We are concerned with support for Computers, Audio-Visual, Learning Technologies, and Networks. Scheduling is supported by the Registrar's Office, and Facilities is supported by (shockingly) Facilities.
  If we only cared about one Service Category, say "Computers," there would be no need to model the "Support Category / Support Unit" concept, because it would be a given – there would only be one.
- Each Support Unit could support one or more Service Categories. E.g., Sam's Call Center provides Tier 1 support for Computers, Audio-Visual, Learning Technologies, and Networking.
- Support for Department-owned rooms is not within the scope of this initiative; support will be provided by the owning Department's Local Support Unit.
- Support for Classrooms (GPC and non-GPCs) or a Room Block of GPCs will be provided by a Support Unit during a Time Block for a Support Level (Tier.) That is, for a given Room Block (available via the Classroom reporting the Incident) for a given Service Category Level (e.g., Computers – Tier 1) during a particular Time Block, a particular Support Unit will provide support. This concept is represented via the "Support Responsibility" concept, an associative entity which indicates the responsibility of a Support Unit to provide support for a Service Category Level for a Room Block during a Time Block. There are three general possibilities:
  1. Support for the Room Block will be provided exclusively by the Local Support Unit (the Department);
     - this only applies to non-General Purpose Classrooms (Department "owned")
  2. Support for the Room Block will be provided exclusively by the Central Support Unit;
     - Will this happen? Is this a goal?
  3. Support for the Room Block) will be provided by the Local Support Unit during "normal business hours" (a Time Block) and by the Central Support Unit outside of "normal business hours."

Classroom Support

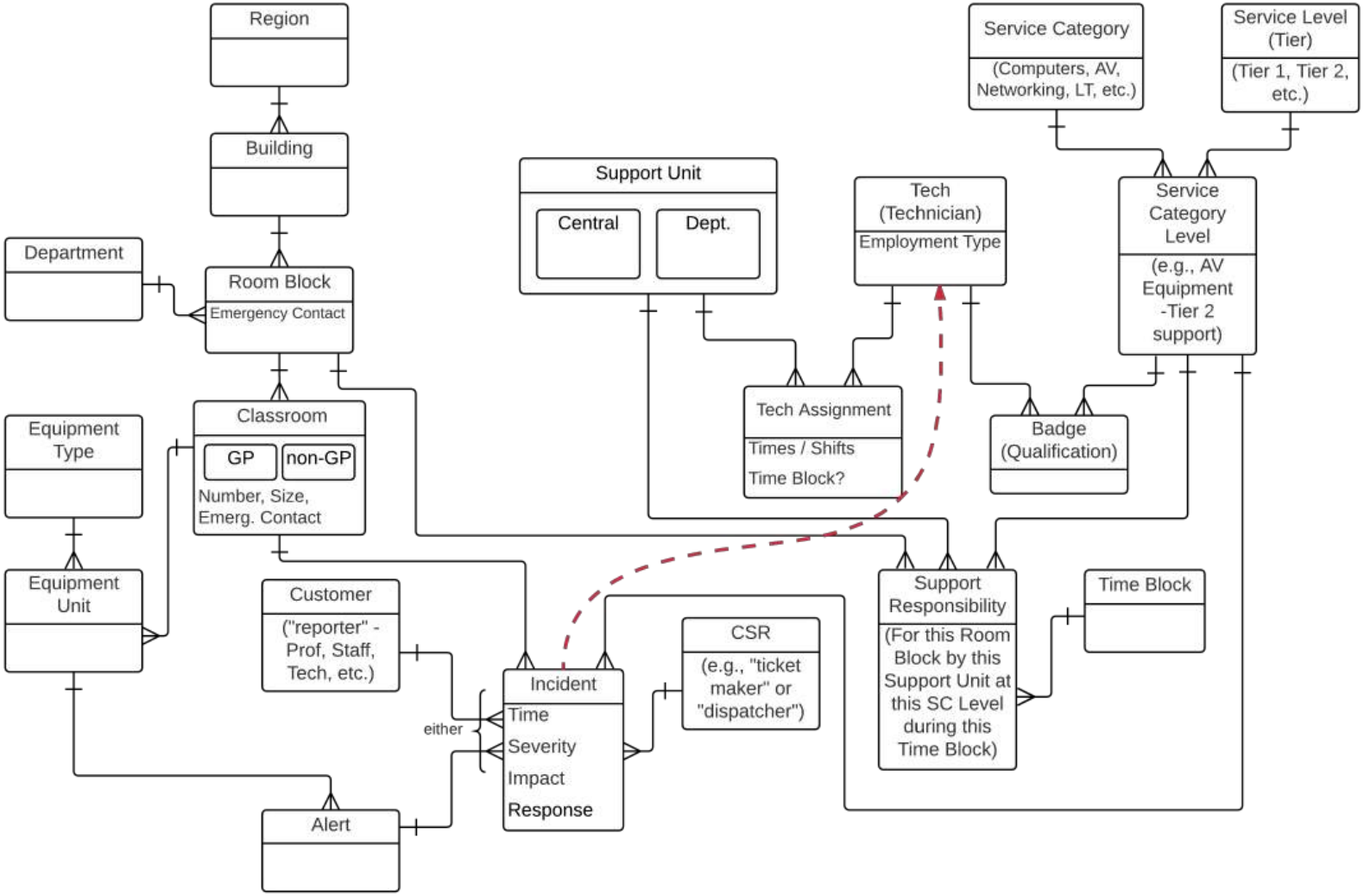
     - Is this the "normal" case?
     - Should it read "after normal business hours?" That is, will Central ever provide support both before and after normal business hours?
- Each Tech may be badged for one or more Service Category Levels, and for each Service Category Level there may be one or more Badged Techs. A M:M relationship.
- Each Tech may be assigned to one or more Support Units during a given time period, and for each Support Unit there may be one or more assigned Techs. A M:M relationship, but will a constraint be that a Tech can only be assigned to one Support Unit at a time?
- An Incident for a particular GP Classroom can be raised by either a Customer (the "reporter" – Faculty, Staff, Tech, …?) or an automated Alert raised by a an Equipment Unit located on a particular GP Classroom.
- The "dispatcher" or "CSR" at Room Support (?) assigns (or routes?) an Incident to the appropriate Support Unit based on the Support Responsibility.

Putting all this to work…

The goal is to automatically route an Incident to one or more Techs. When an Incident is raised, Dispatch will always create a Ticket, and then route it to the appropriate Tech(s) based on Service Category Level (Service Category and Service Level,) Time Block, Room, and Support Unit. Here's how…

- When an Incident is raised, we know the Room Block (via Room,) the Time Block, and the Service Category Level, therefore we know the Support Responsibility, and therefore the Support Unit.
- We also know which Techs are badged for that Service Category Level, and which Techs are assigned to that Support Unit at that time.
- Now we have a pool of Techs the Incident could be dispatched to, for them to "bid on," Uber-style.

The underlying "Concept Plus" Model



Summary of findings

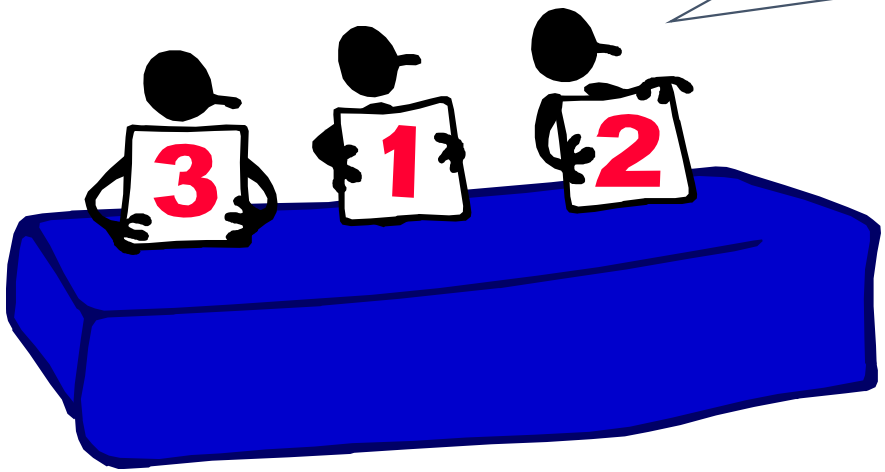
The assertions and the ERD showed the idea could be implemented:

- When an Incident is raised, we know the Room Block (via Room,) the Time Block, and the Service Category Level, therefore we know the Support Responsibility, and therefore the Support Unit.

- We also know which Techs are badged for that Service Category Level, and which Techs are assigned to that Support Unit at that time.

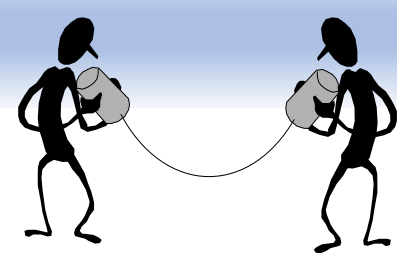
- Now we have a pool of Techs the Incident could be dispatched to for them to “bid on.”

Example – Data Modelling as the basis for COTS configuration*



The beginning of the end?
Various commentators on my data modelling career, mid-1990s

Redemption, via a SAP implementation



The client...

Could you come on over and do that thing you do?

That entity data stuff with the boxes and lines

We're implementing something called SAP. Our CEO told us to!

When you did this with our Work Order Management system, we all felt we understood our business better than we ever had.

They say it's a terrible idea, a waste of time, and will you *please stay home!*

Alec...

I guess.
What thing in particular?

Oh, data modelling.
Sure - what's the project?

Ah… sounds familiar.
How can I help?

Nice. And what do your SAP consultants say about me coming out?

I'm on my way.

The outcome – using DM for ERP configuration

The situation:
- Manufacturer selects SAP as platform for process transformation
- Desire to understand as-is *business processes* to map to package and decide on configuration options
- Client felt the integrator was coercing them, wanted my help

The #1 reason for unhappiness with the selected COTS solution – *a data model mismatch!*

The approach:
- Team of 7 builds 45 entity *concept model* over two days
- Identify "what's good, what's not good" about current business rules, revise concept model
- Use this knowledge on configuration activities with concept model as an overall map

Vendor
Country
Site
Plant
Plant Location
Equipment Item & Type
PO, PO Line Item
Req'n, Req'n Line Item
Release, Release Line Item
Work Definition, WD Line Item
etc. etc. etc.

The key points:
- ***Client-initiated, not IT***
- Now a global showcase account
- Client – "More value from those two days than anything else we did!"
- Me – "I'm not irrelevant!"

"Quick wins" example – selecting an application with verbs and nouns

Selecting of new Financials app is completely stalled despite huge effort to develop and maintain a BDM*

Requirements	D&B	Oracle	SAP	Coda	etc.
1	Y	Y	Y	Y	
2	Y	Y	Y	N	
3	Y	Y	Y	Y	
4	N	Y	N	Y	
5	N	N	Y	Y	
6	Y	Y	Y	Y	
7	Y	Y	Y	Y	
8	Y	Y	Y	Y	
9	Y	N	Y	N	
10	N	Y	N	Y	
11	Y	Y	Y	Y	
12	Y	Y	Y	Y	
13	Y	N	Y	Y	
14	Y	Y	N	N	
…					
…					
858	N	N	N	Y	
859	Y	Y	Y	Y	

*** Big Dumb Matrix**

BDM issues

- Time consuming
- Most apps meet most criteria
- Still can't tell if an app will work well in your environment

Using DM for purchased application selection – verbs and nouns

The problem:

- Selection of new Financials app is hopelessly bogged down (and a matrix of almost 1000 "requirements" wasn't helping)
- Worse – *matrix points to the app no one wants!*

The approach:

- Small team builds "thing model" (concept model, ~60 entities total, 15 "core")
- For each core entity, identify 3 to 5 life cycle events
- For each event, develop scenario w. data
- Turn over to ***paid*** app vendors – "Show us!"
  - "How do you support the data model?"
  - "How do you handle scenarios?"

"Things we track…"

- **Project, Work Order**
- **Plant, Plant Equipment**
- **Product Type, Product Lot**
- **Product Inventory**
- **Sale, Transfer**
- **Location, Ledger Entity**
- **Financial Category**
- **Responsibility Center**
- **Account, Sub-Account**
- **Fixed Asset**

Events that happen to them…"

Fixed Asset is

- **Acquired or Constructed**
- **Depreciated**
- **Transferred**
- **Disposed Of**

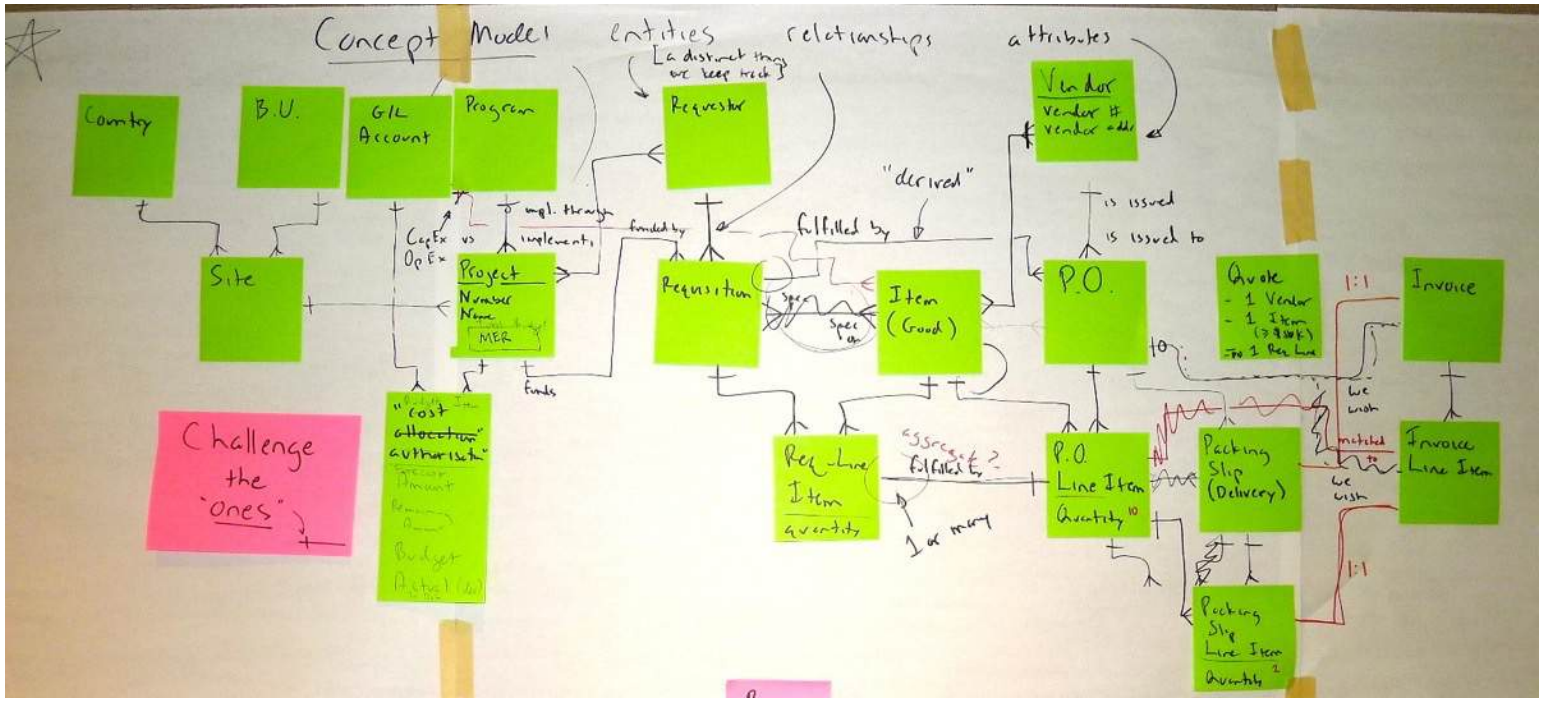
The key points:

- It worked! – saw how an app would support the business
- Didn't initially call it "data modelling"
- Left vendor some room - "Here's how we'd do it."

Another example – Concept Model shows possibility of major process change

Global mining company hires me to help with Business Process in support of ERP changeover.

I "snuck in" some quick, informal Concept Modelling.



This highlighted many areas lacking clarity:

- Program vs. Project
- Site vs. BU Location vs. Country
- Requisition vs. Quote vs. Purchase Order
- The 1:1 relationships among PO/PO Line Item, Packing Slip/Packing Slip Item, and Invoice/Invoice Line Item showed that Invoiceless Payment, a major process change, was possible

I did not use any data modelling terminology until the end!

SW failure if you ignore the process and the data…

U.S. University implementing cloud-based Human Resources and Payroll systems from *the same vendor.*

- Total spend US$80M, nothing salvageable
- University leadership unamused
- I was brought in for “project recovery”

The situation

My assignment –
take a large team through a process model
and data model-based approach –
run 4-day offsite in “The Capsule”
(we felt like astronauts)

A "Futuro" house by
Finnish architect Matti Suuronen

What we learned:

- Little time on “business process”
  - very generic / unrecognisable as “what we do”
  - team tires of this
- Zero time on “data” (no “concept model”)
- Management: "Get on with it – the vendor has seen it all before."
- 100+ programmers begin detailed configuration of *application rules and logic – “Straight to task.”*

Initial focus – too much on "requirements"

Over 100 developers coded detailed business rules and contract terms *separately* into

- Payroll Application
- HR Application

Note: university had over 35 labour unions with complex payroll and benefits policies/rules – ***no rethinking whatsoever!***

Remediation – focus on process and data

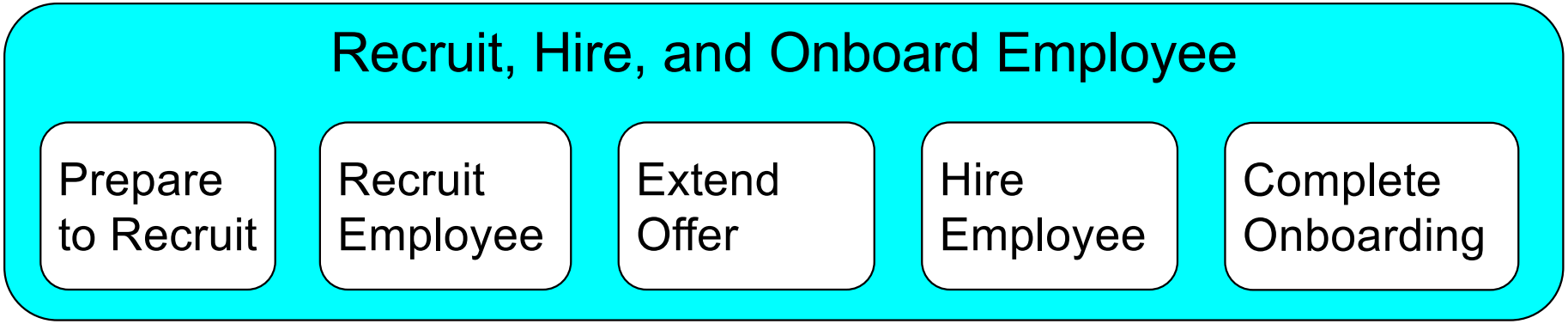
Identified, modelled, analysed, redesigned significant process – “Recruit, Hire, and Onboard Employee," the Case was "Tenure-Track Faculty”

- Developed *scope model* (invaluable!)
- Developed augmented scope model
- Assessed and redesigned based on “what”
- Added “who & how” to create a to-be *augmented scope model*

Modelled seven critical concepts in data – “what do we mean by…”

- Supervisory-Organisational Hierarchy
- Position-Based Management
- Visible Application Workflow
- etc.

First, identify main phases in a Scope Model



Augmented Scope Model ("what") for the full process

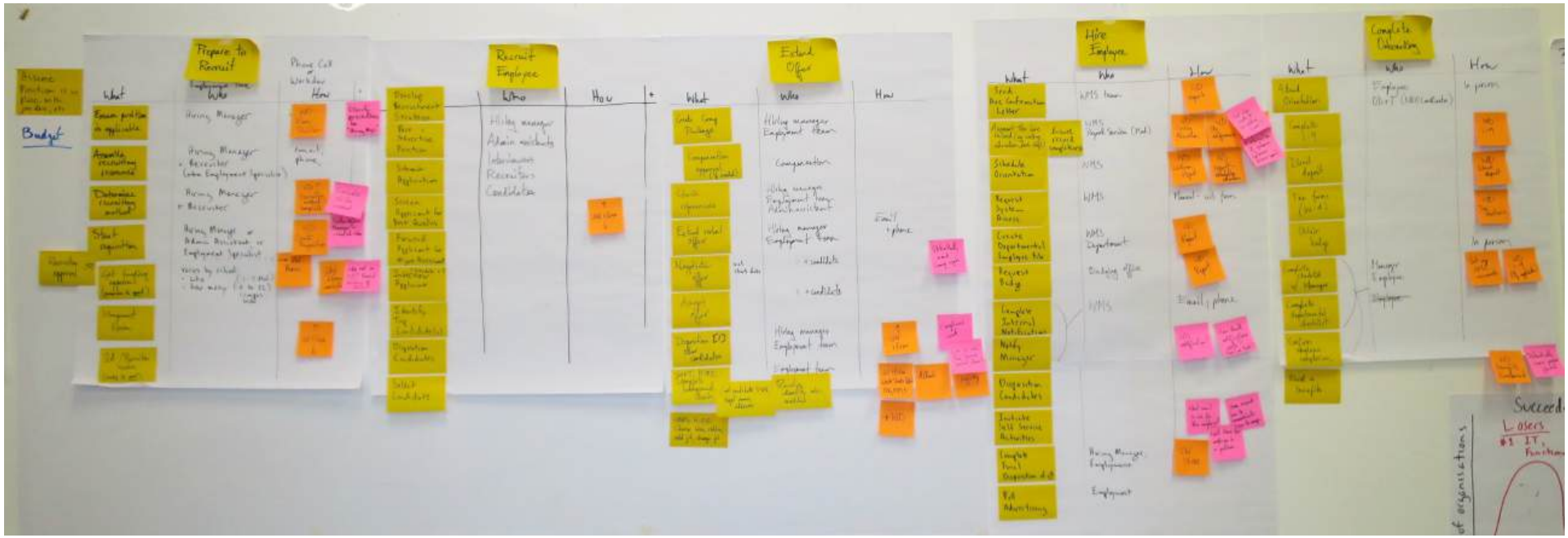
Recruit, Hire, and Onboard Employee

- *Identify* Payroll Title & Job Title
- *Develop* Job Description
- *Obtain* Funding, Compensation, & Space Approvals
- *Identify* Supervising Manager
- *Determine* Recruiting Method
- *Assemble* Recruiting Resources (e.g., search committee)

Active verbs & nouns

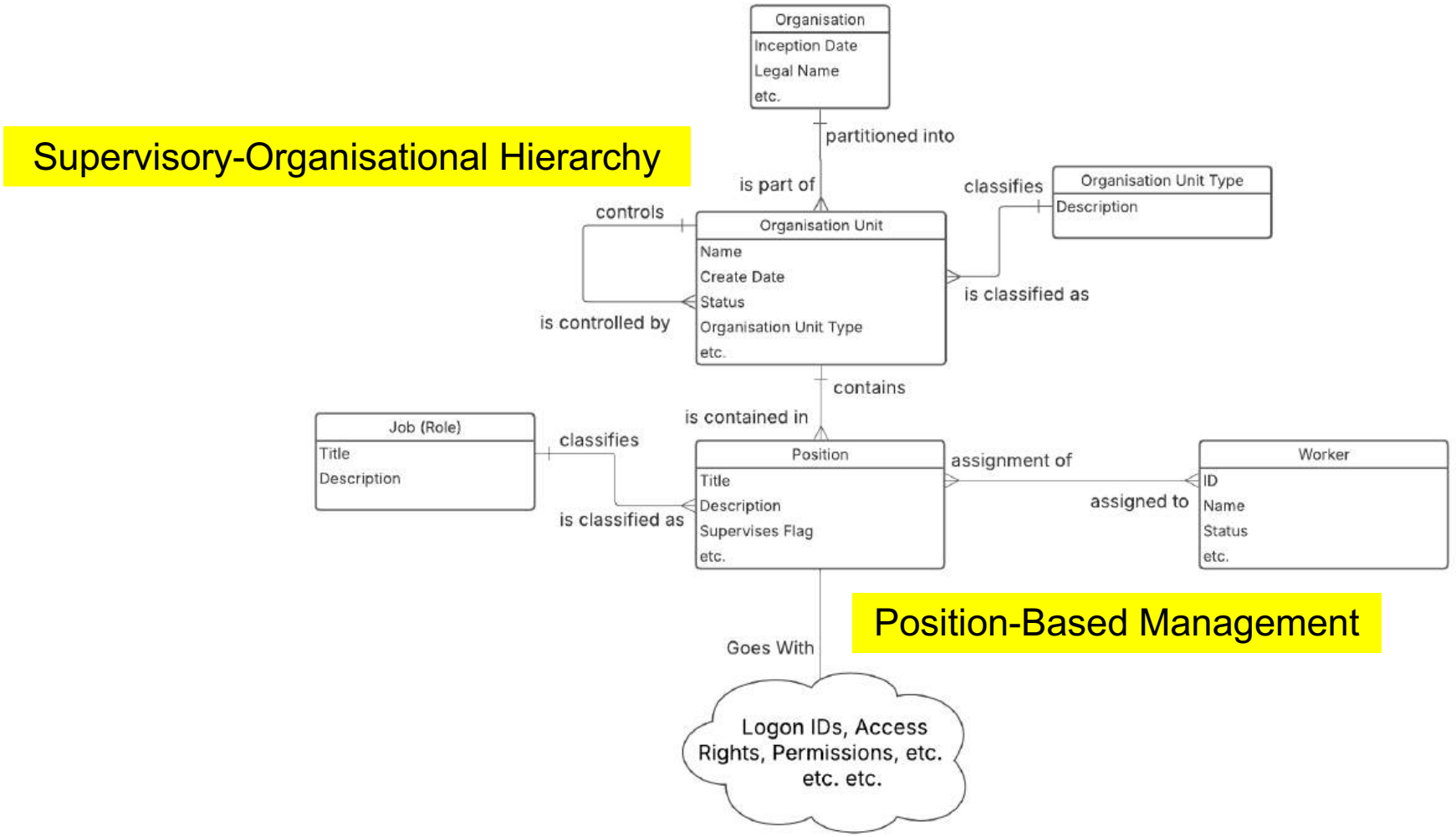
- *For the first time,* the entire end-to-end process is visible
- A surprise to *everyone* how much work it is, and *how many functions participate!*
- Still no reference to “who or how,” just "what" – *active verb* + noun
- *This is critical to build support for change – it “depersonalises” in a good way!*

Then add "who and how"

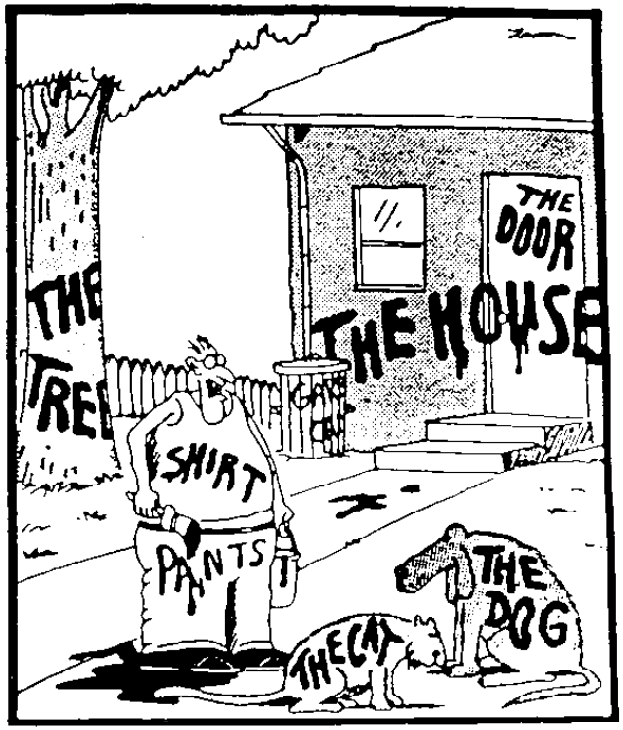


Next, add “who” (which role) and
“how” (which tool or system function) and "notes."
Now we have the basics of a to-be process design, and *an understanding of which steps will be supported by which system functions – great for understanding if the COTS app will actually work!*
(And easier than jumping into detailed requirements and flow modelling.)

And of course, Concept Modelling was <u>really</u> important



Remember, it all starts with language!



"Now! *That* should clear up a few things around here!"

We hope you enjoyed the class!

1. Requirements Definition
   - Goals, Issues, and the Return of Modelling
   - Case Study - Integrating the Techniques
2. Business Process Fundamentals
   - Five Things You Need to Know
   - Discovering, Scoping, & Assessing Your Processes
3. Concept Modelling Fundamentals
   - E, R, A - A Concept Model's Essential Components
   - Drawing Your Model for Maximum Understanding
4. Business Process Workflow Modelling & Design
   - Five Core Guidelines for Great Swimlane Diagrams
   - Facilitating a Process Mapping Session
   - Assessment of the As-Is and Transition to the To-Be
5. The Process-Data Connection
   - The Natural Synergy between Process & Data Models
   - Process-Data Synergies in Modelling, Analysis, & SW

Other courses for analysts by Alec Sharp

Working With Business Processes – *Process Change in Agile Timeframes* 2 days

Business processes matter, because business processes are how value is delivered. Understanding how to work with business processes is now a core skill for business analysts, process and application architects, functional area managers, and even corporate executives. But too often, material on the topic either floats around in generalities and familiar case studies, or descends rapidly into technical details and incomprehensible models. This workshop is different – in a practical way, it shows how to discover and scope a business process, clarify its context, model its workflow with progressive detail, assess it, and and transition to the design of a new process by determining, verifying, and documenting its essential characteristics. Everything is backed up with real-world examples, and clear, repeatable guidelines.

Business-Oriented Data Modelling – *Useful Models in Agile Timeframes* 2 days

Data modelling was often seen as a technical exercise, but is now known to be essential to other initiatives such as business process change, requirements specification, Agile development, and even big data, analytics, and data lake implementation. Why? – because it ensures a common understanding of the things – the entities or business objects – that processes, applications, and analytics deal with. This workshop introduces concept modelling from a non-technical perspective, provides tips and guidelines for the analyst, and explores entity-relationship modelling at contextual, conceptual, and logical levels using techniques that maximise client involvement.

Working With Business Processes Masterclass – *Aligning Process Work with Strategic, Organisational, and Cultural Factors* 3 days

This 3-day interactive workshop combines the core content from two highly-rated classes by Alec Sharp – "Working With Business Processes" and "Advanced Business Process Techniques." This structure is popular because it gets both new and experienced practitioners to the same baseline on Claritiq's unique, agile, and ultra-practical approach to Business Process Change. First, it shows how to effectively communicate Business Process concepts, discover and scope a business process, assess it and establish goals, and model it with progressive detail. Then, it shifts to advanced topics – specific, repeatable techniques for developing a process architecture, encouraging support for change, and completing a feature-based process design. The emphasis is always on ensuring business process initiatives are aligned with human, social, cultural, and political factors, and enterprise mission, strategy, goals, and objectives.

Business-Oriented Data Modelling Masterclass – *Balancing Engagement, Agility, and Complexity* 3 days

Our most popular workshop! This intensive 3-day workshop combines the core content from two popular offerings by Alec Sharp – "Business Oriented Data Modelling" and "Advanced Data Modelling." First, the workshop gets both new and experienced modellers to the same baseline on terminology, conventions, and Clariteq's unique, business-engaging approach. We ensure a common understanding of what a data model *really* is, and maximising its relevance. Then, we provide intense, hands-on practice with more advanced situations, such as the enforcement of complex business rules, handling recurring patterns, satisfying regulatory requirements to model time and history, capturing complex changes and corrections, and integrating with dimensional modelling. Always, the philosophy is that a data model is a description of a business, not of a database, and the emphasis is on engaging the business and improving communication.

Model-Driven Business Analysis Techniques – *Proven Techniques for Processes, Applications, and Data* 3 days

Simple, list-based techniques are fine as a starting point, but only with more rigorous techniques will a complete set of requirements emerge, and those requirements must then be synthesised into a cohesive view of the desired to-be state. This three-day workshop shows how to accomplish that with an integrated, model-driven framework comprising process workflow models, a unique form of use cases, service specifications, and business-friendly data models. This distinctive approach has succeeded on projects of all types because it is "do-able" by analysts, relevant to business subject matter experts, and useful to developers. It distills the material from Clariteq's three, two-day workshops on process, data, and use cases & services.

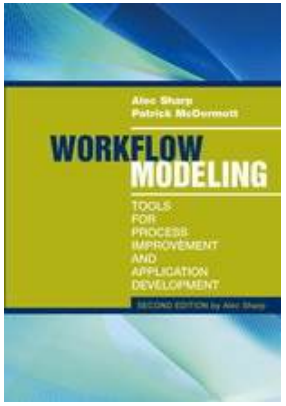
**** Note: two-day in-person workshops are delivered virtually as three half-day sessions via Zoom.*
Three-day in-person workshops are delivered virtually as five half-day sessions via Zoom.

Thank you – stay in touch!

CLARITEQ

Alec Sharp
Clariteq Systems Consulting Ltd.
West Vancouver, BC, Canada

- asharp@clariteq.com
- ig: @alecsharp01
- www.clariteq.com
- http://amzn.to/dHun1o



And most of all, if you have questions or comments…
don't be shy – send me a note!