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

Presented by Adept Events and Clariteq Systems Consulting Ltd.
for NL Ministerie van Defensie
25 november 2024, Den Haag NL

Alec Sharp
Senior Consultant, Clariteq
West Vancouver, Canada
asharp@clariteq.com

Ministerie van Defensie

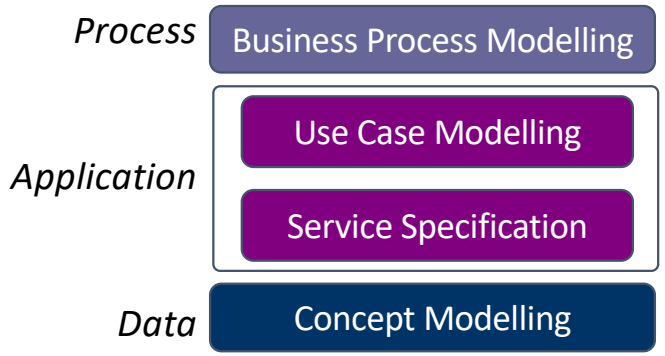
///AdeptEvents

Presenter background…

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

- 40+ years experience as an independent consultant:
  - Business Process Change – discover, model, analyse, and design/redesign processes
  - Application Requirements Specification
  - Concept Modelling / Data Management

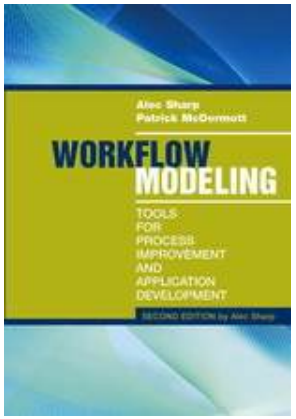
    +
  - Facilitation & Organisational Change
  - Project Recovery



- Consulting, teaching, speaking globally

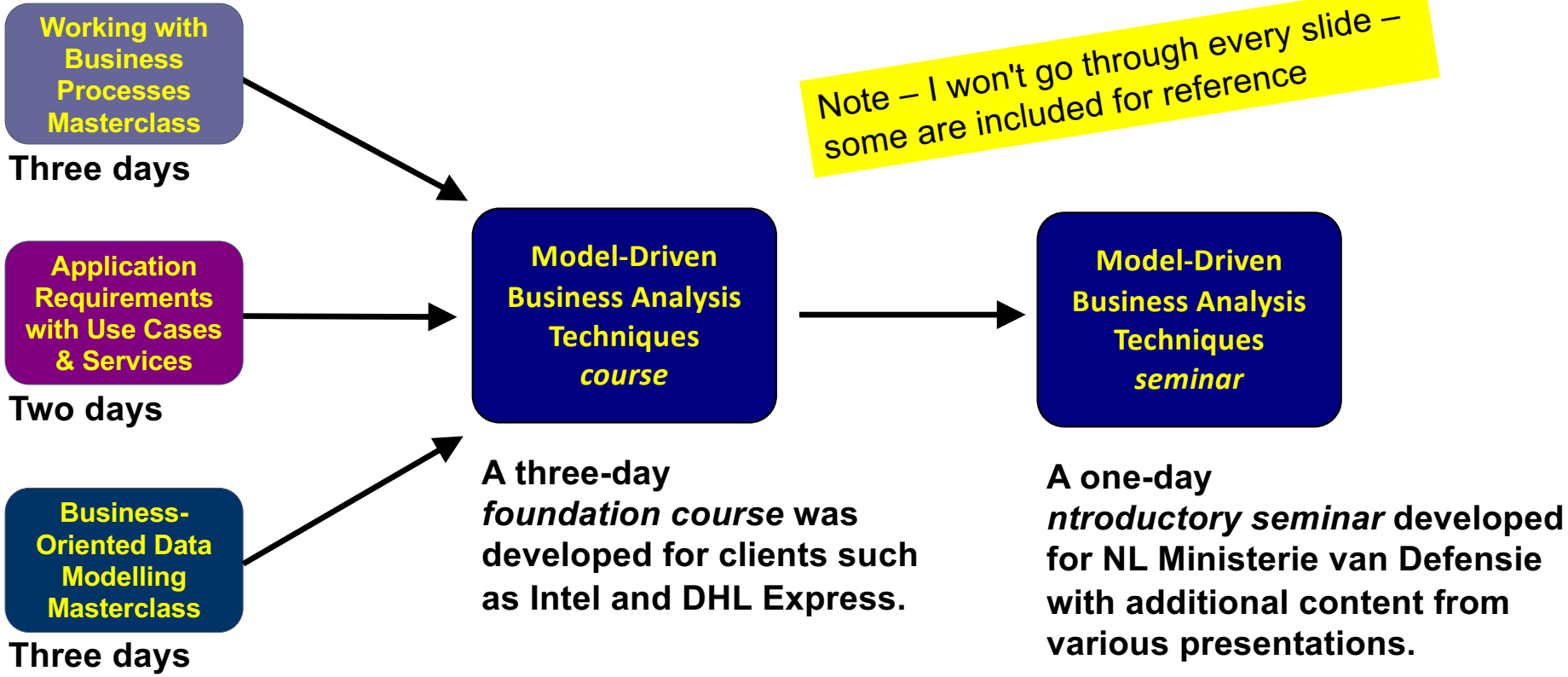
- Awarded DAMA's global Professional Achievement Award for contributions to "human-friendly" data modelling

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



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

Realistic expectations – an introduction, not a deep dive



Logistics

Schedule (CET):

- start 09:00
- class (90 minutes) 09:00 – 10:30
- break (15 minutes) 10:30 – 10:45
- class (105 minutes) 10:45 – 12:30
- lunch (60 minutes) 12:30 – 13:30
- class (90 minutes) 13:30 – 15:00
- break (15 minutes) 15:00 – 15:15
- end 17:00

We will also have some short nature breaks and email breaks.

Lots of content for a one day seminar!

- Smaller discussion-based exercises – no major case study, but some hands-on

- We will also rely on me demonstrating techniques, discussion, etc.
  Questions?
  Please *feel free to speak up!*

Overview of topics

A Business Analysis framework

Concept Modelling as a Foundation

Business Process Essentials

The Data-Process Connection

(Optional) A little more on Use Cases & Services

"Business Analysis" gets criticised because of the extremes

Simplistic methods at one extreme: can do as much harm as good	The goal lies in the middle ground:	Overly complex methods at the other extreme: difficult for businesspeople to verify

List-form requirements, typically a Business Requirements Document – lacking *context*

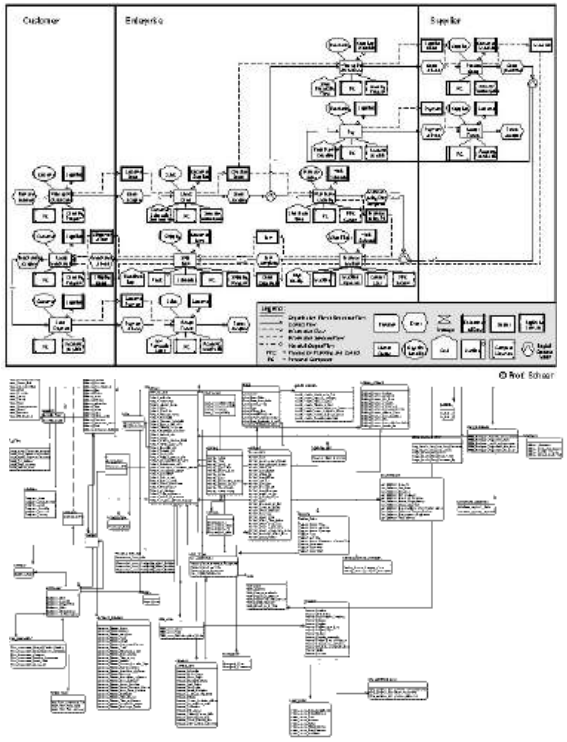
ID#	Business Feature	Requirement Type	Business Unit(s) Affected	Potential Application(s) Impacted
BRQ025	files that are available for the selected day.		Readiness	
OMSPI-BRQ026	System shall include all outage status in the Transmission Outage report.	Core	Operation Readiness	WebOMS
OMSPI-BRQ027	System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

Client – *understandable, and therefore verifiable.*

Analyst – *doable, within Agile timeframes.*

Developer – *unambiguous, complete, actionable*

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



The problem 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, …*

What they are 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

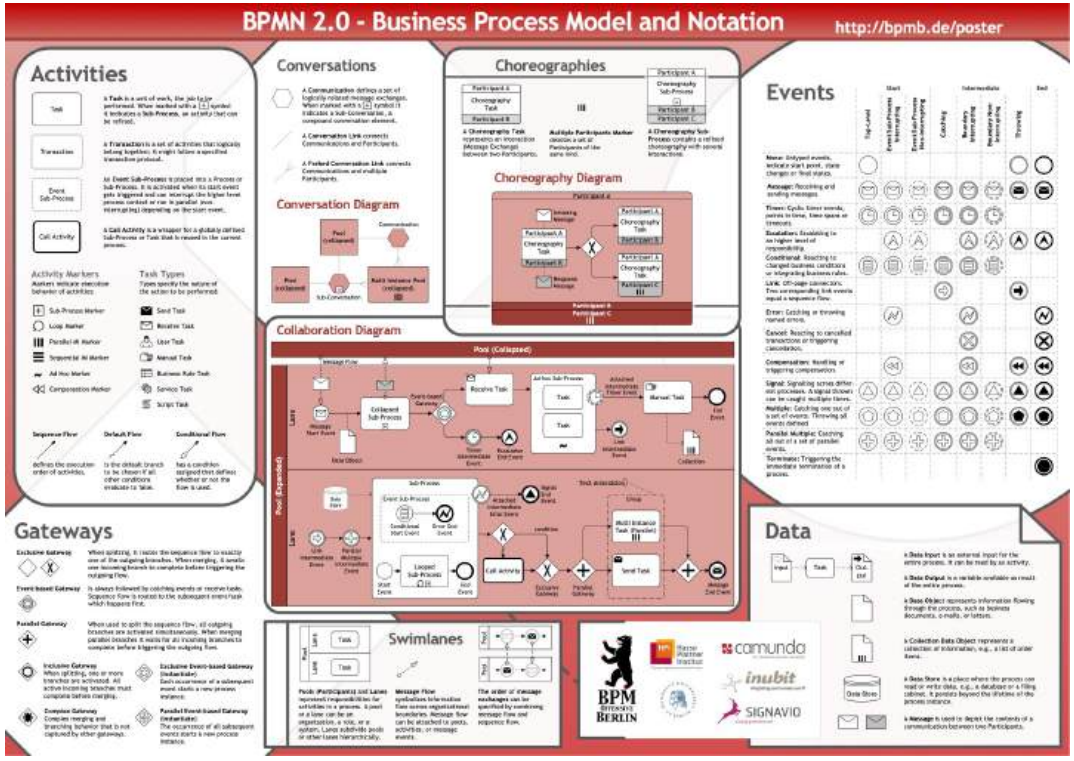
- *Business Process Models* to show where this fits in the end-to-end process
- *Business Object Models* to show the required information

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.”

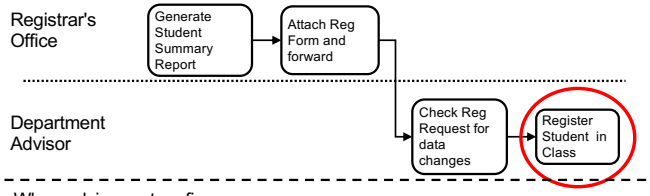
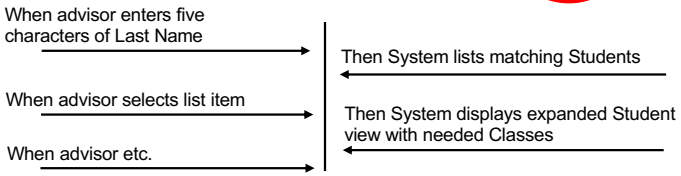
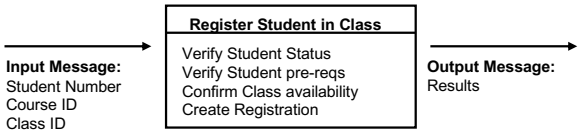
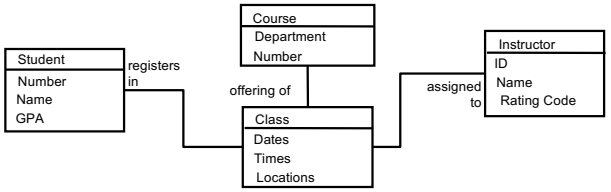
From the original UML specification: “The Unified Modeling Language (UML) is a graphical language for visualizing, specifying, constructing, and documenting the *artifacts of a software-intensive system*.”

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



Best Practice – a business-friendly, model-based framework for Business Analysis

The Clariteq Framework for Business Analysis

Framework Layer		Technique sample	What it covers
Goals	*Business Objectives*	The university is initiating the "Strategic Enrollment" program to raise Student graduation rates in part by ensuring Classes are available for Student registration when needed.	✓ ***Project Charter*** – documents the rationale, objectives, scope, and success measures for the project
Process	*Business Process*		✓ ***Process Model*** - shows "what" in a Scope Model, then "who & how" in a Workflow Model – the steps done by the actors in the process
Application	*Presentation Services (user interface)*		✓ ***Use Case*** – models how an actor interacts with a system to obtain (trigger) a service, typically to complete a step in a process
	Business Services (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	*Data Mgmt. Services (databases)*		✓ ***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*

Overview of topics

A Business Analysis framework

Concept Modelling as a Foundation

Business Process Essentials

The Data-Process Connection

(Optional) A little more on Use Cases & Services

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

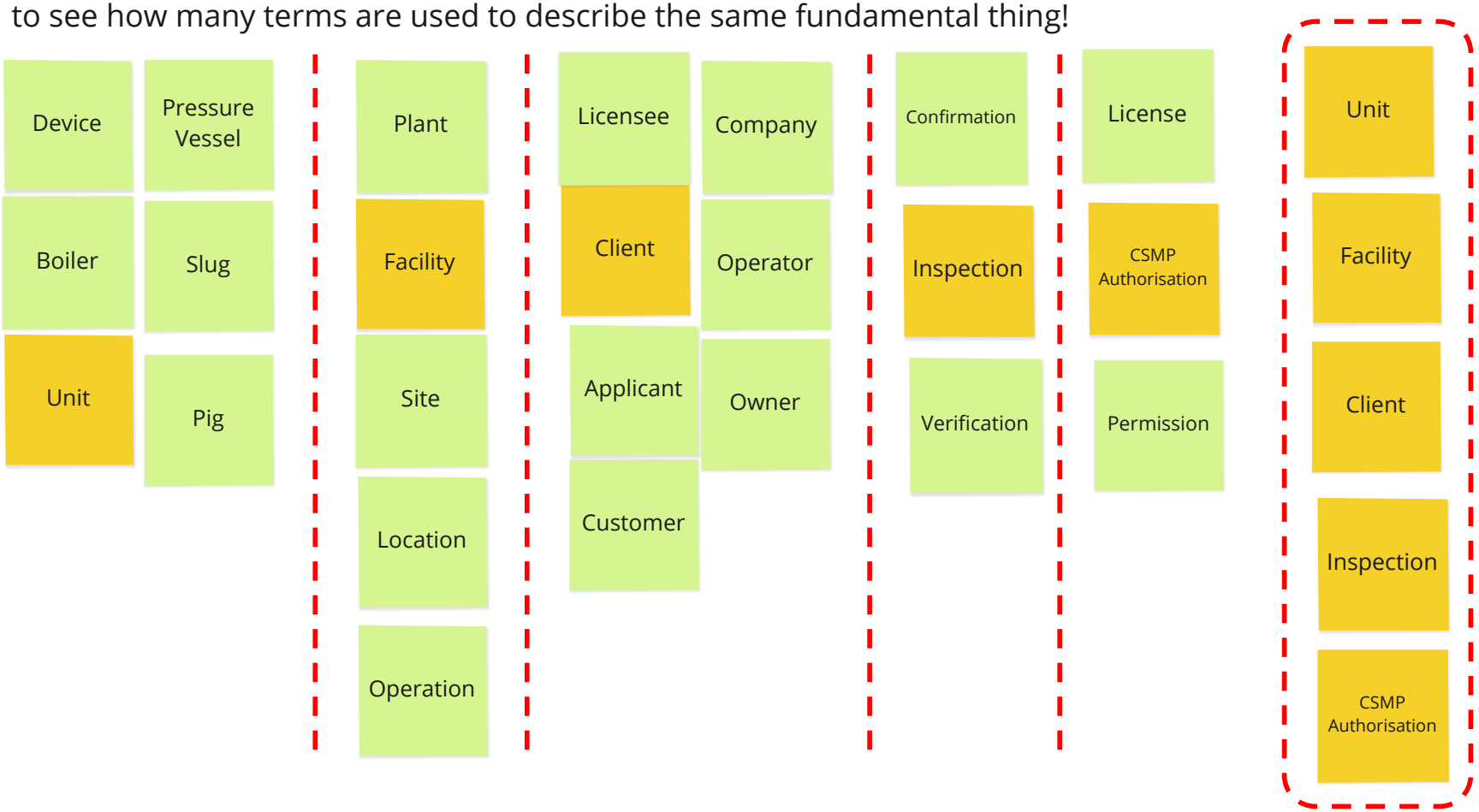
Always start with terminology (the "things")

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?

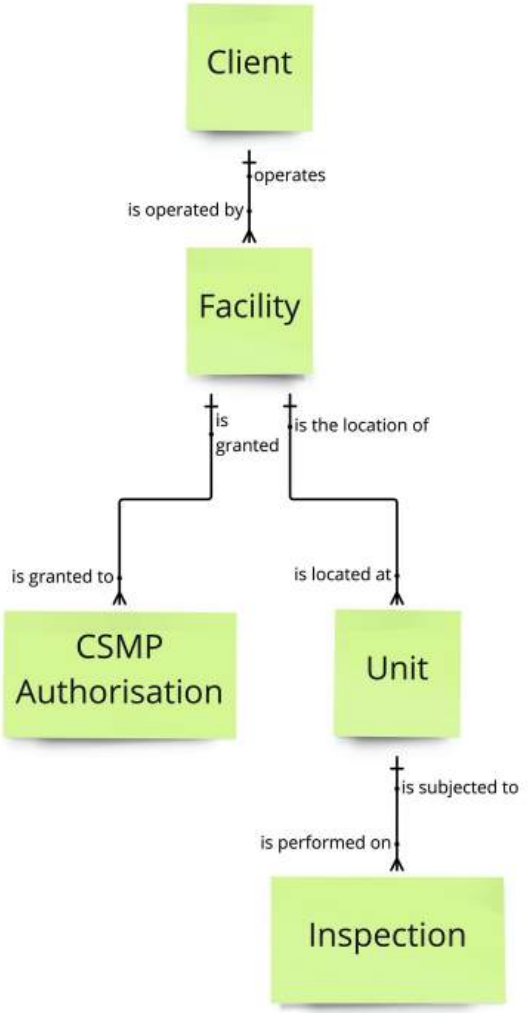
Review of a Miro example – Terminology Analysis

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!



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

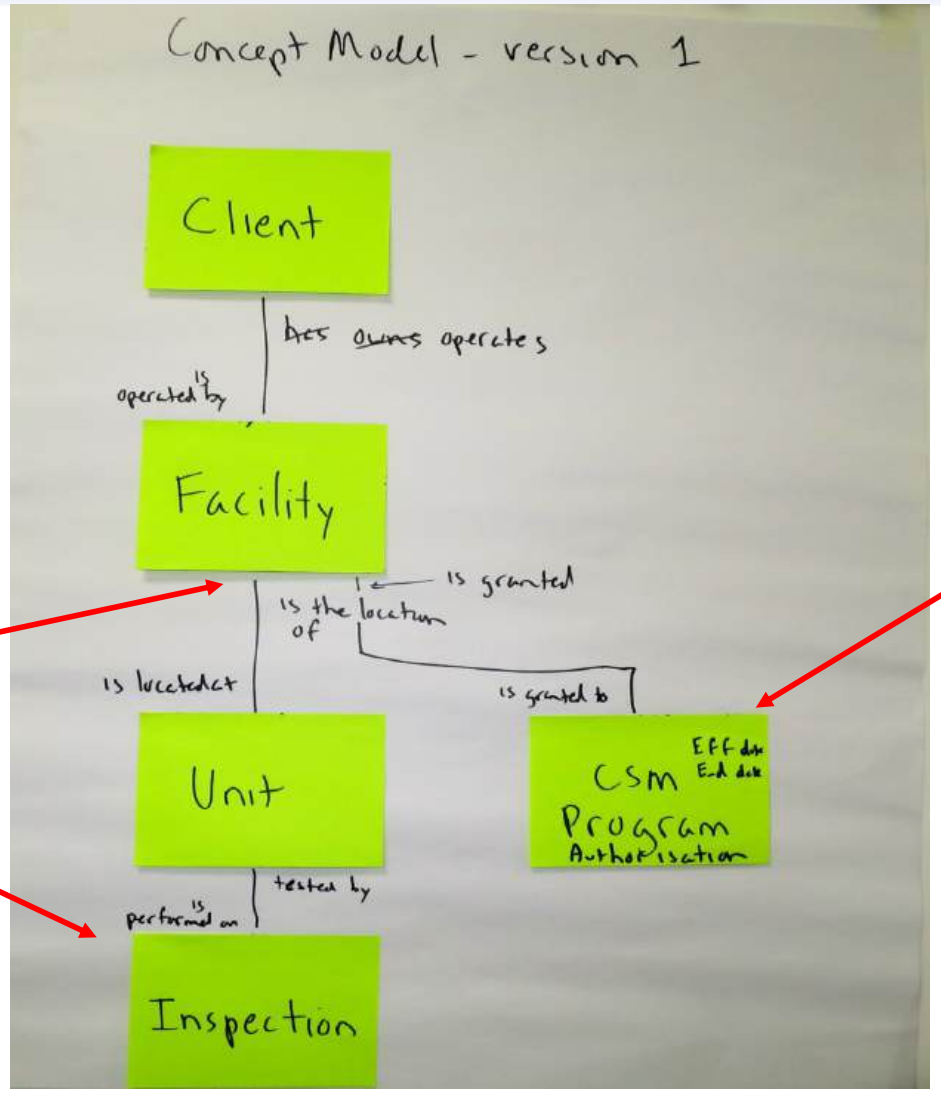
1. We arranged the entities / business objects by dependency
2. Then we drew relationship lines
3. Then we added a relationship name in each direction
4. Only then did we state (in words) the cardinality (1:1, 1:M, M:M) and then update the diagram with hash marks (⫞) and crowsfeet (⋔)



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 ___

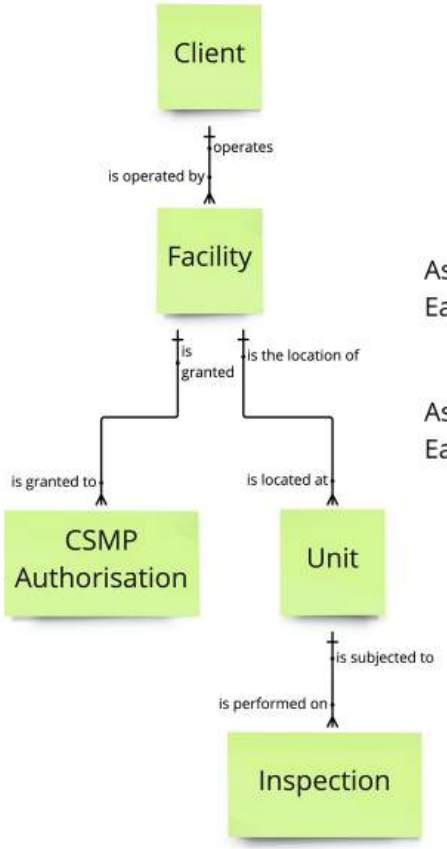
Assertion:
Each Facility is the location of ___

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.

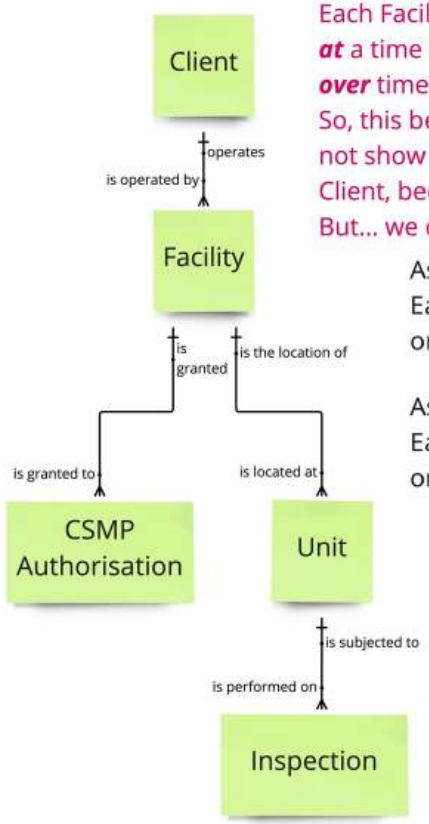
Assertion:
Each Client operates
one or more Facilities

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

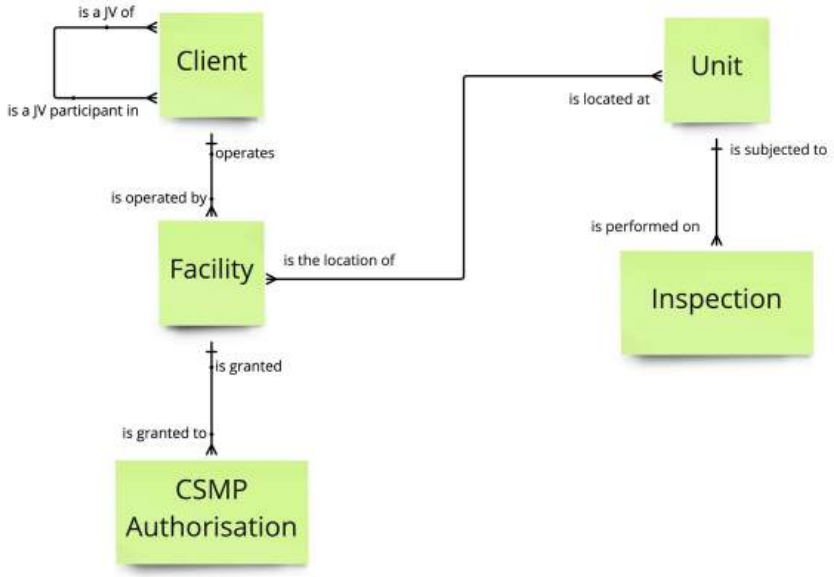
Assertion:
Each Unit is located at
one Facility

Assertion:
Each Unit is subjected to
one or more Inspections

Assertion:
Each Inspection is performed on
one Unit

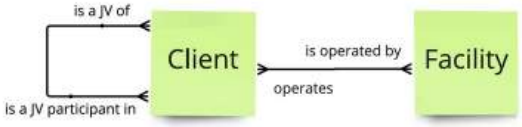
Concept Model Version 2; revised from challenging Assertions

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

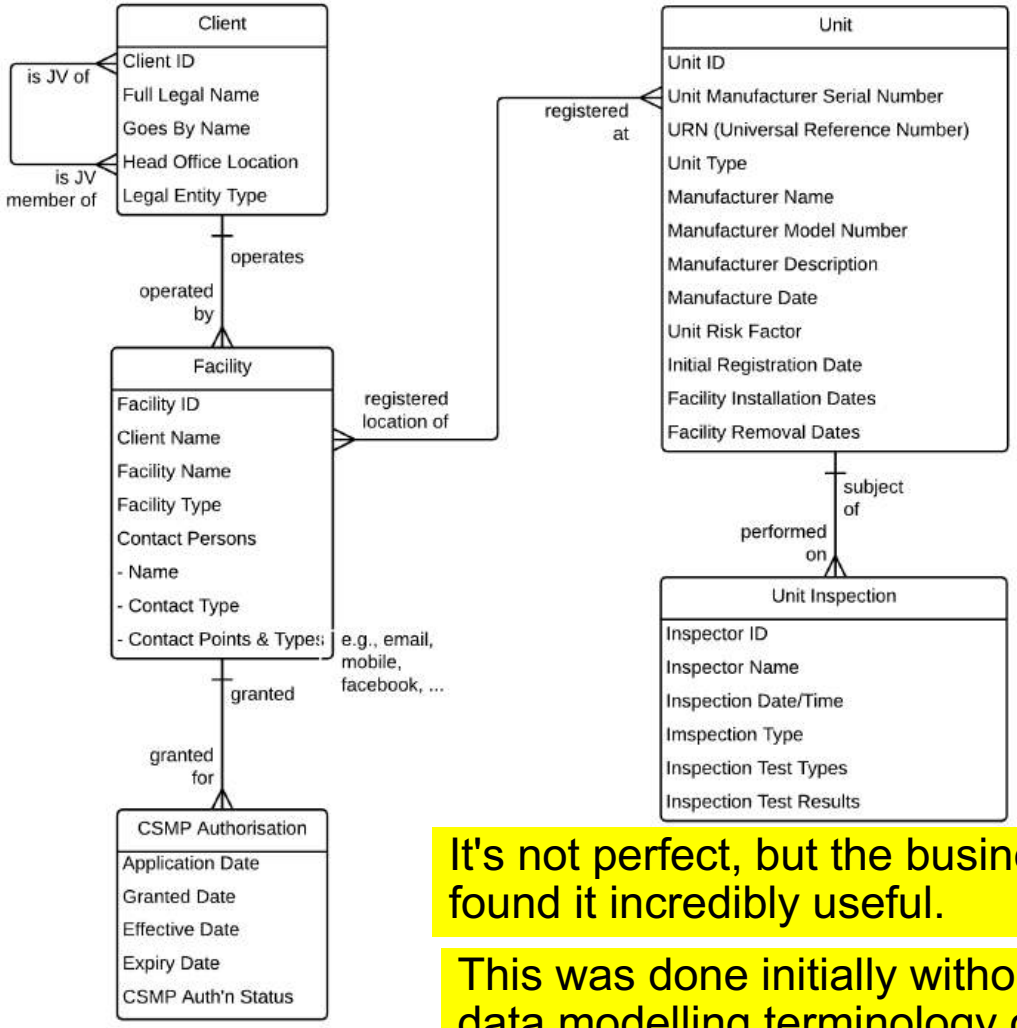


Note:
You don't always get what you *want* or what you think is the *right* thing in Concept Modelling. In this case the client (the Regulator) said they always wanted a Facility to be operated by ONE AND ONLY ONE Client.
If a Facility was operated by multiple Clients, they would require the Clients 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!

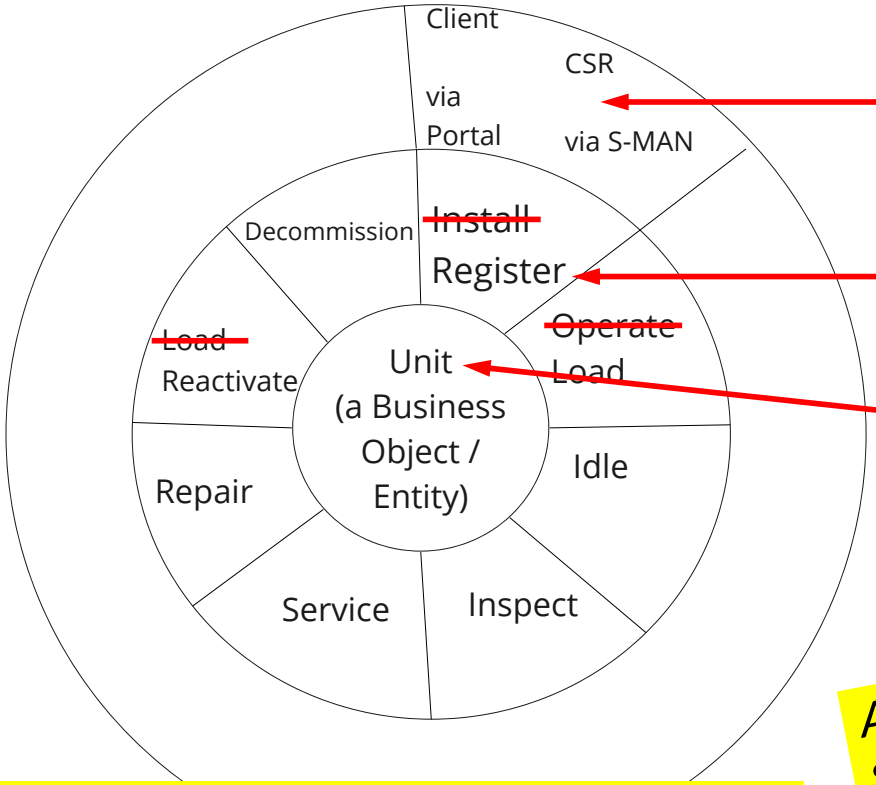
Model took ~90 minutes

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?

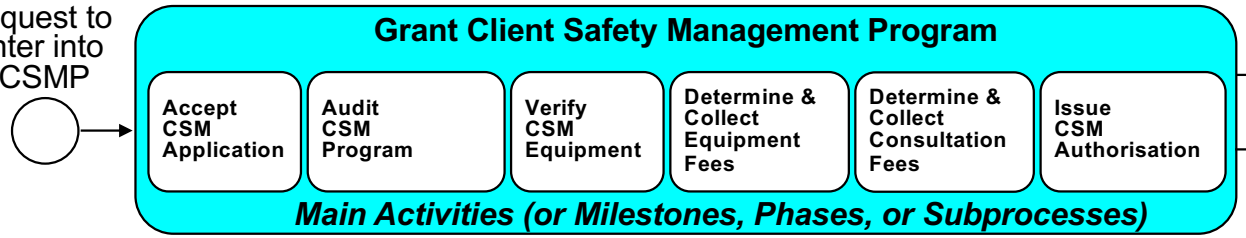
Use Case	*Use Case or User Story - add Who and How*
Service Specification (Events)	*Service (or Event) - add a Verb to the Noun*
Concept Model	*Entity or simply a "thing" - a core Noun*

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

Supports *Service-Oriented Business Analysis*

Clarify scope of the new process and identify participants

Trigger:
Client submits request to enter into a CSMP



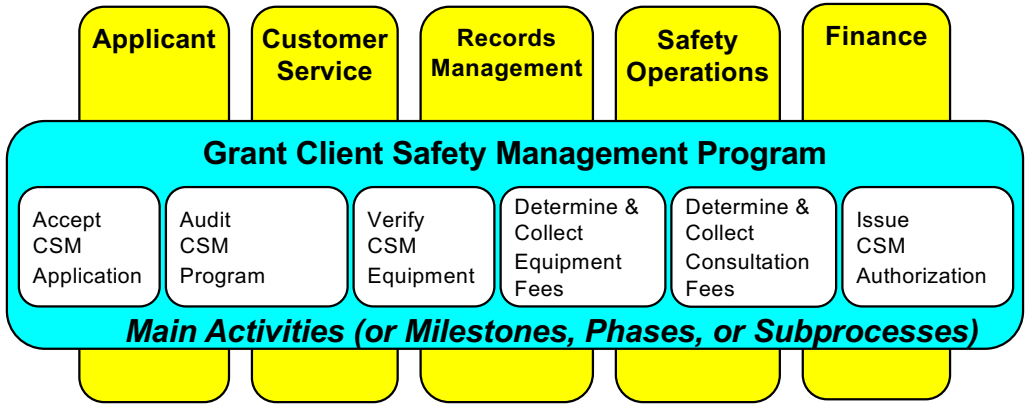
Client Result:
Approval granted for a self-managed safety program.

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

Cases:

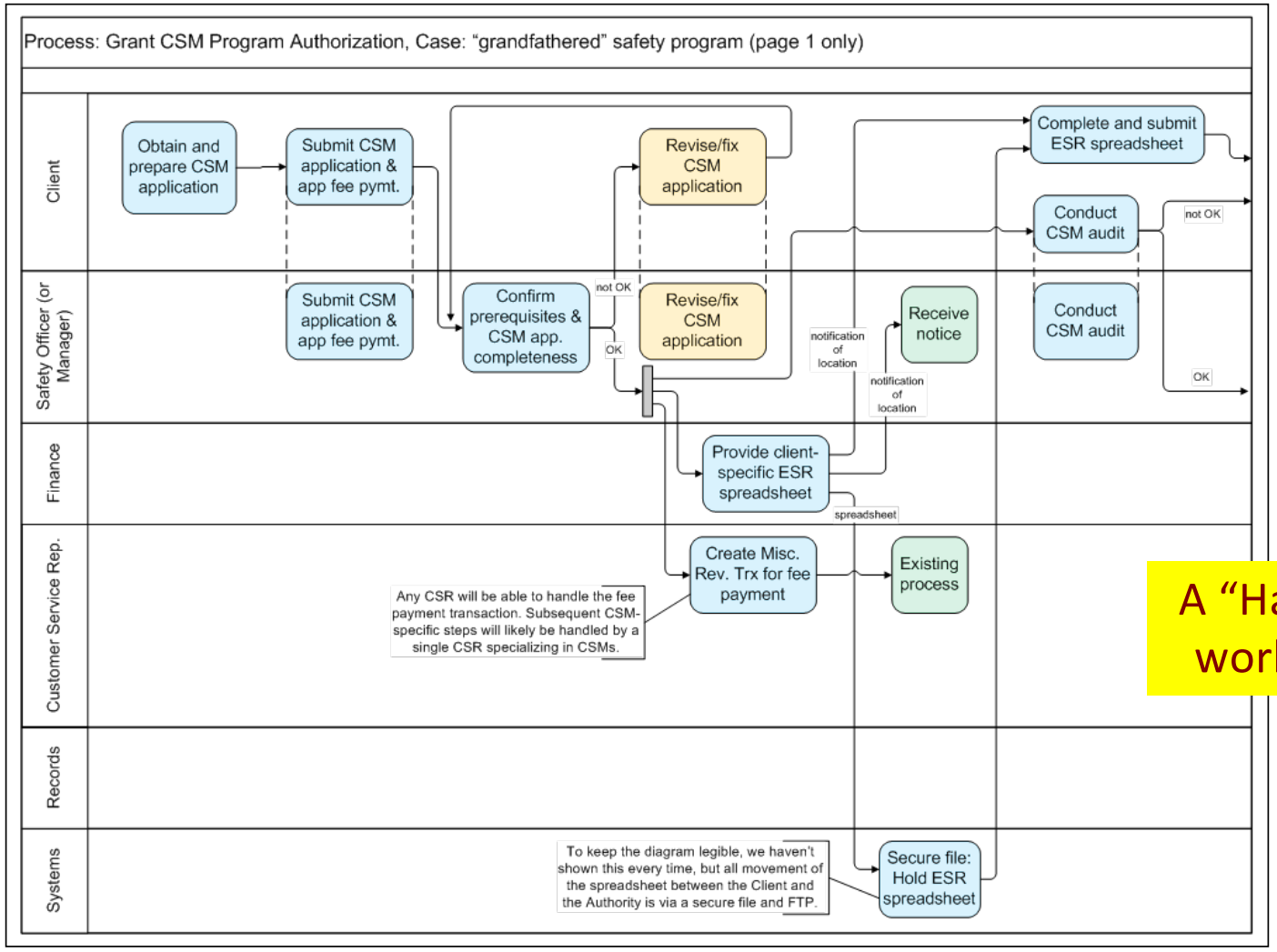
- New
- Grandfathered
- Ownership Change

Process Scope Model – pure "what"…



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

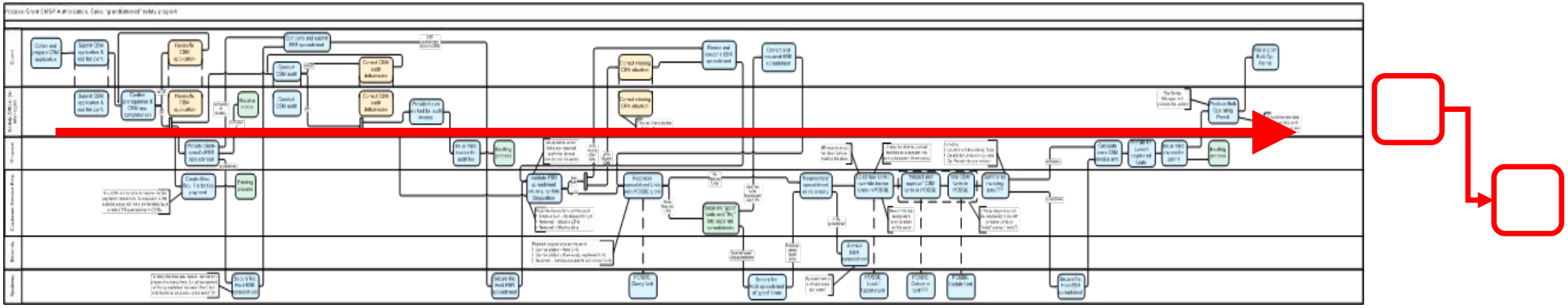
The initial, business-friendly workflow model



A "Handoff Level" workflow model

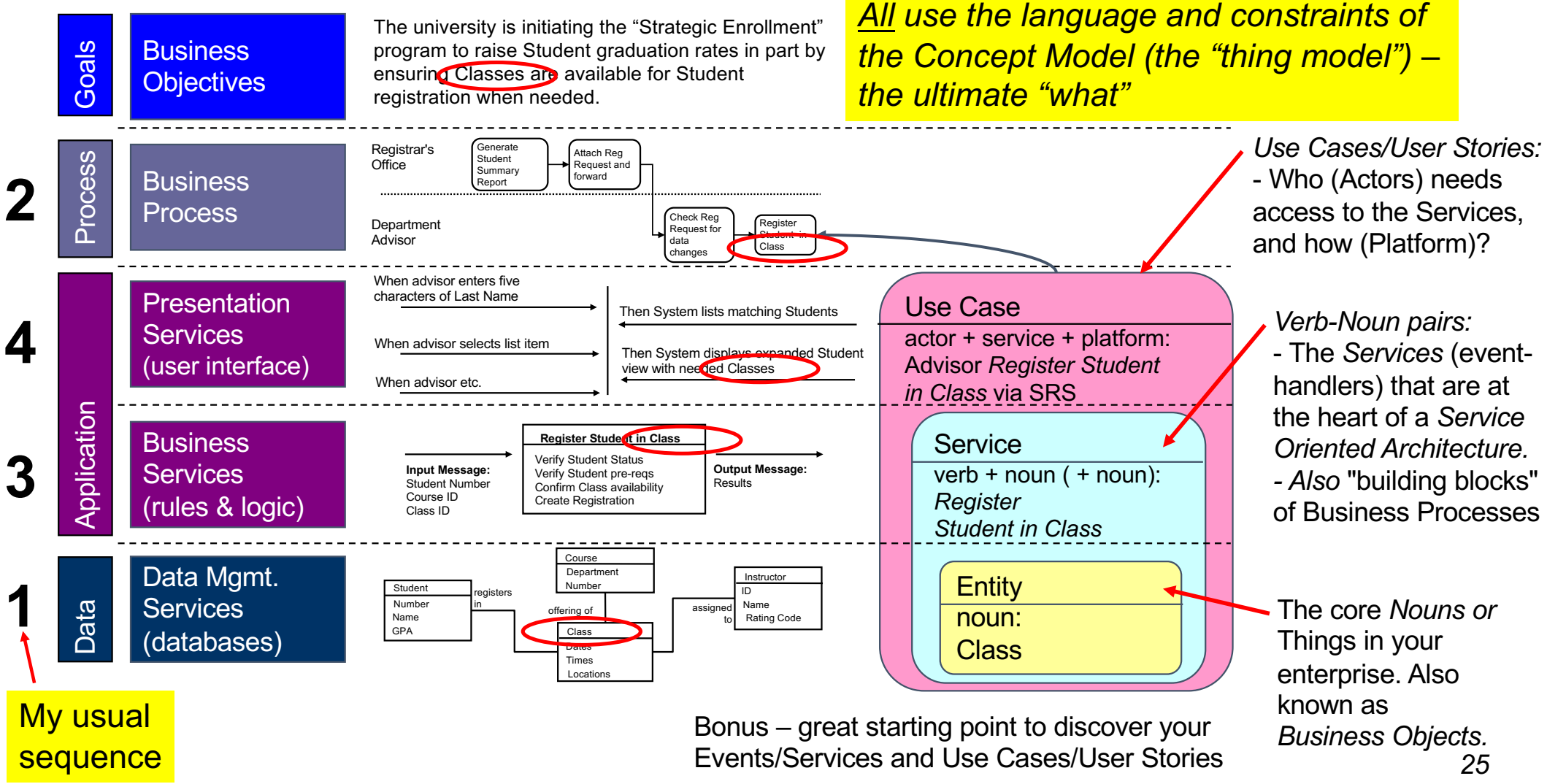
Mission accomplished! Conclusions:

- "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*



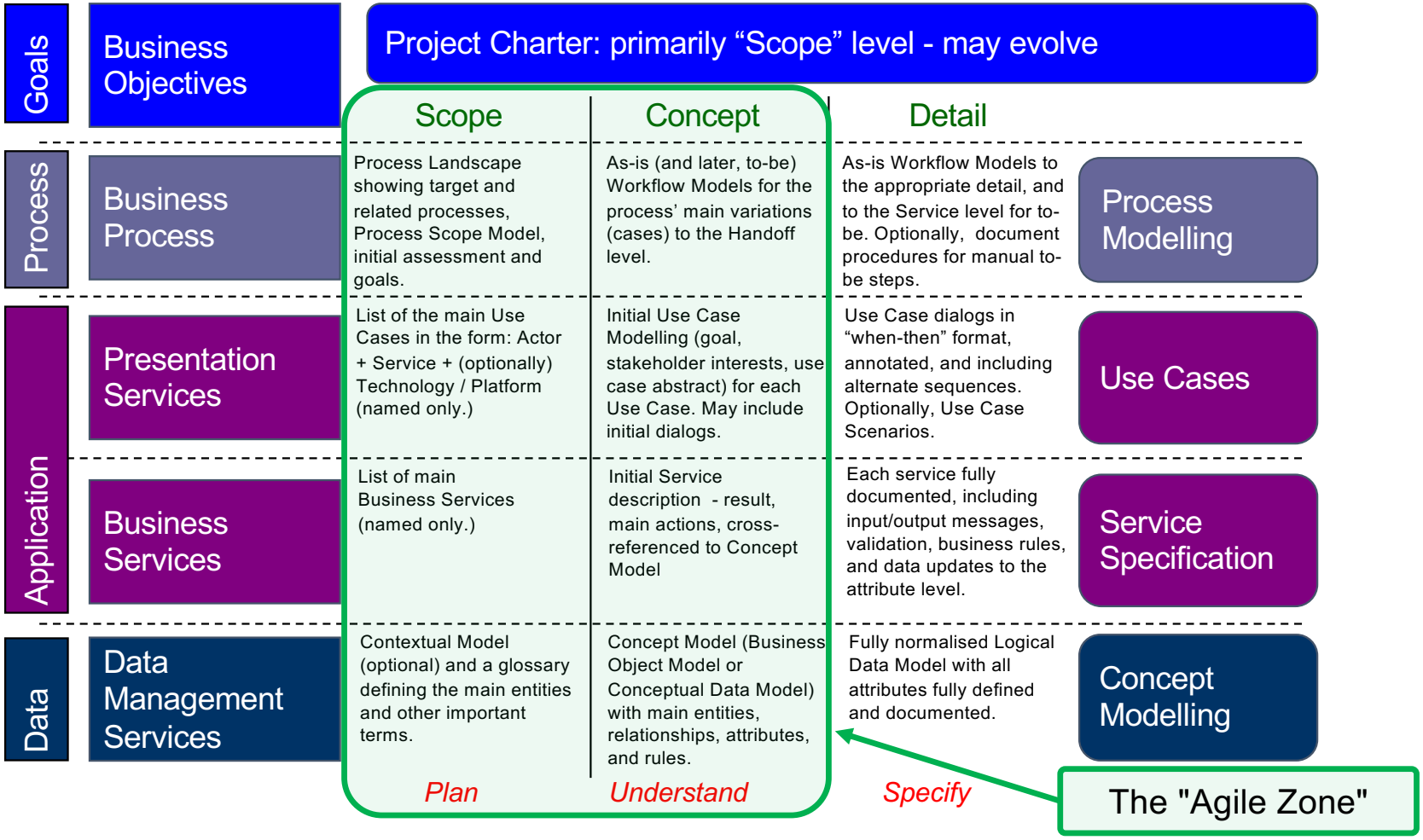
1. If requirements, issues, assumptions, etc. are in lists, people will argue endlessly; if they are in an *integrated* and *understandable* set of models, it’s much harder to dismiss the reality of the situation
2. Process Models, Use Cases, Service Specs, & *Concept Models*: ***essential!***

Key point! Everything relies on the Concept Model



Concept Modelling supports Agile

Clariteq framework for analysis and architecture



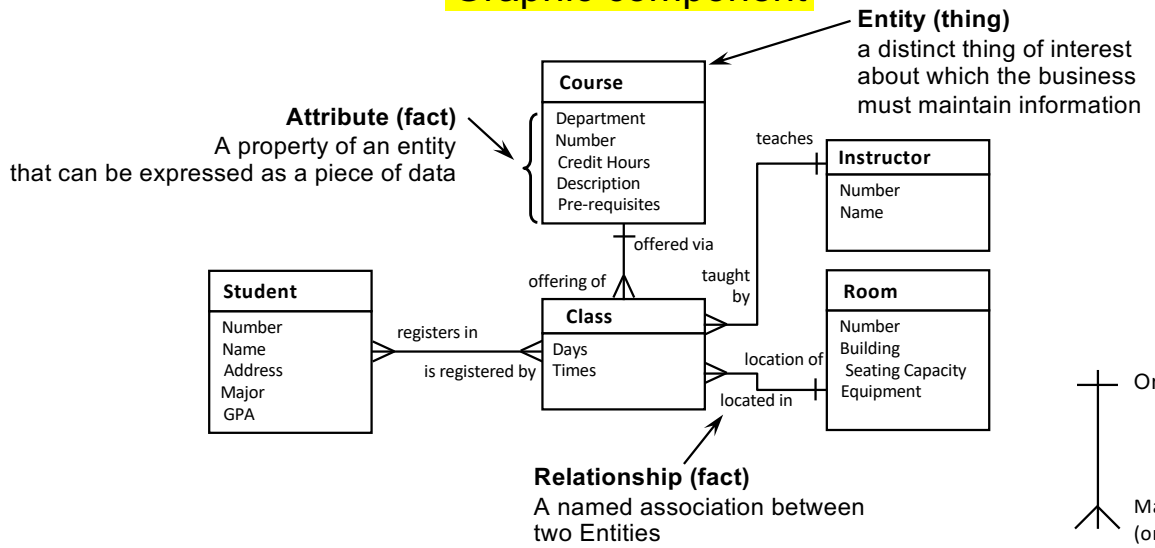
What is a Concept Model / Business Object Model / Domain Model…?

- 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!

Narrative component

Graphic component



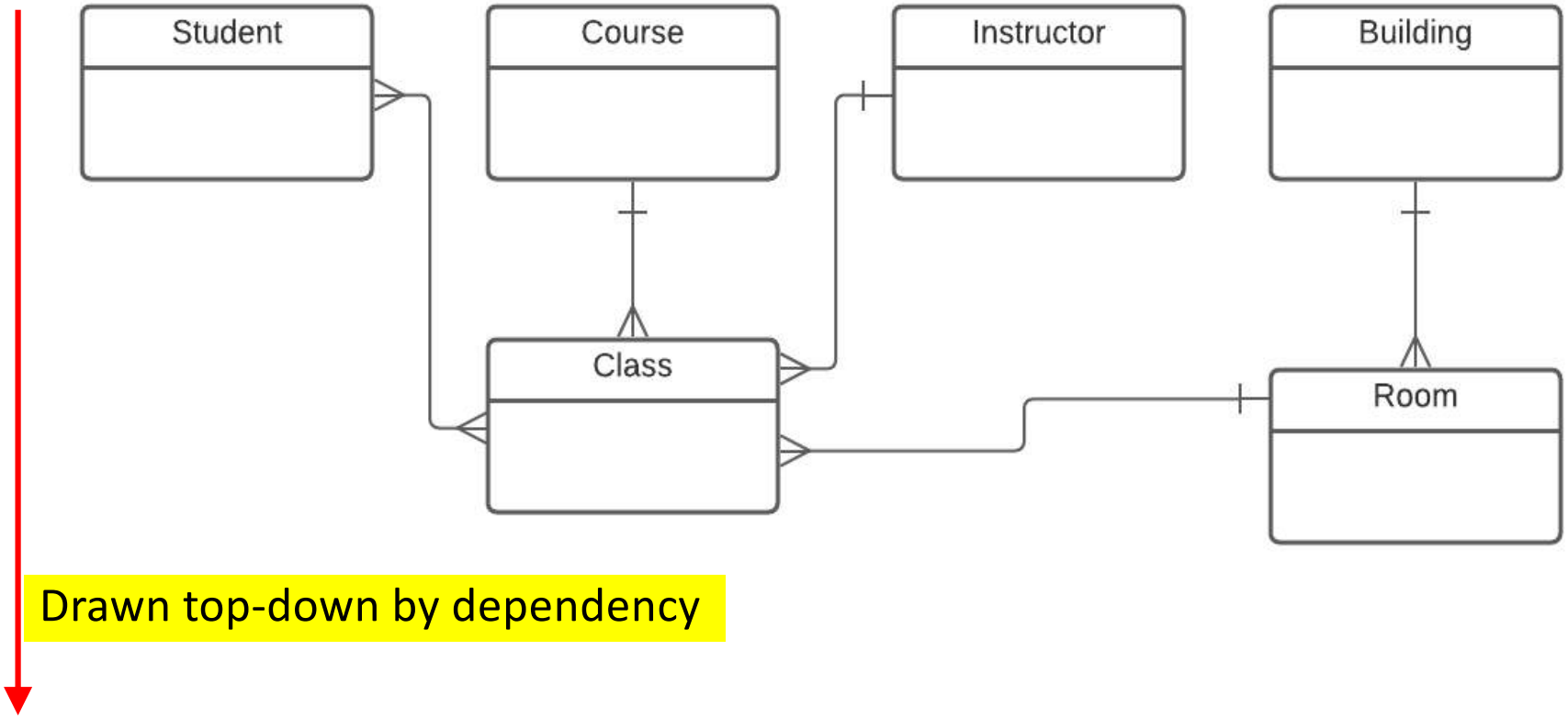
Student definition:
A Student is any person who has been admitted to the University, has accepted, and has enrolled in a course within a designated time. Faculty and staff members may also be Students

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

Independent Entities at the top



Drawn top-down by dependency

The basics: ERA – *Entities*

A distinct thing about which the enterprise must maintain facts in order to operate.

Criteria –

- *singular noun* – we can talk about *one of them* ("Employee," not "Staff")
- *multiple* instances
- must *need to* and be *able to* keep track of *each* instance
- has *facts* (attributes & relationships) that must be recorded
- makes sense in a *"verb-noun"* pair
- *NOT* an *artifact* like a spreadsheet or report

Must be:

- named:
  business-oriented noun / noun phrase
- defined:
  "What is one of these things?" or
  "What do you mean by _____?"

Two basic types:

- independent – can stand alone
- dependent – must have one or more parents

~~Staff~~

Employee

"person"

~~Locations~~

Site

"place"

~~Inventory~~

Part

"thing"

~~Weddings~~

Marriage

"event"

~~Laws~~

Regulation

"concept"

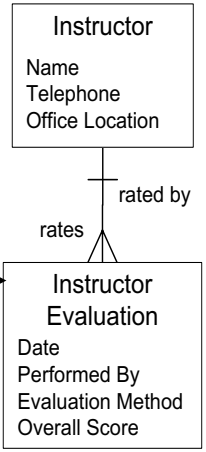
~~Corporate Structure~~

Department

"organisation"

Independent
- "strong"
- no relationships "on top"
  (no parents)

Dependent
- "weak"
- one or more relationships
  "on top" (to parent(s))

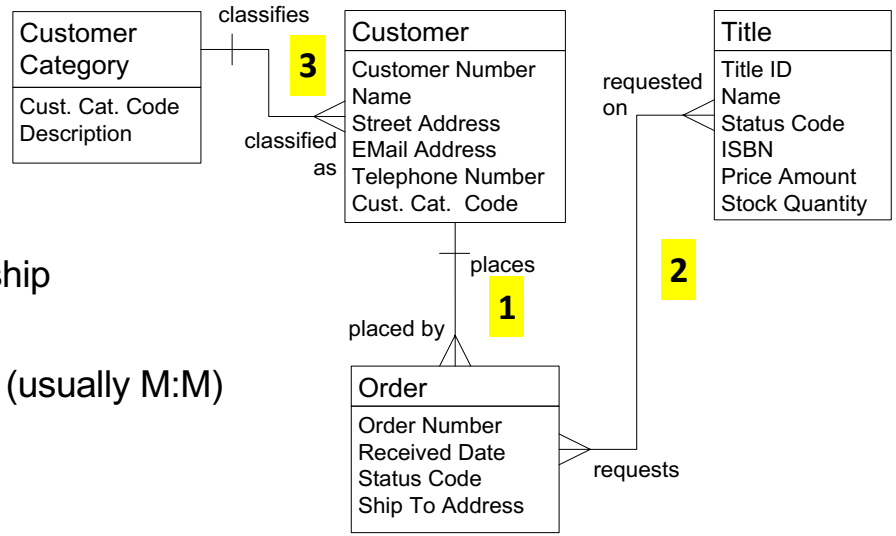


The basics – ERA – *Relationships*

An association between Entities that the business must keep track of

Named in both directions

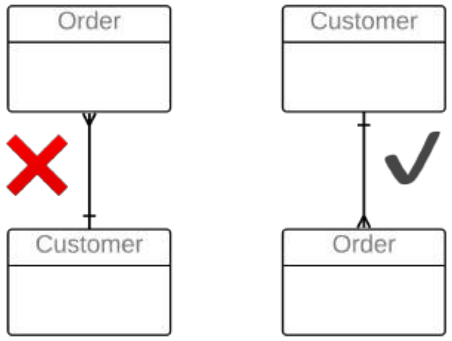
- verb-based phrase
- the name tells us *how* they are related,



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)
- *always* state relationships as *assertions* and *challenge* them!

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

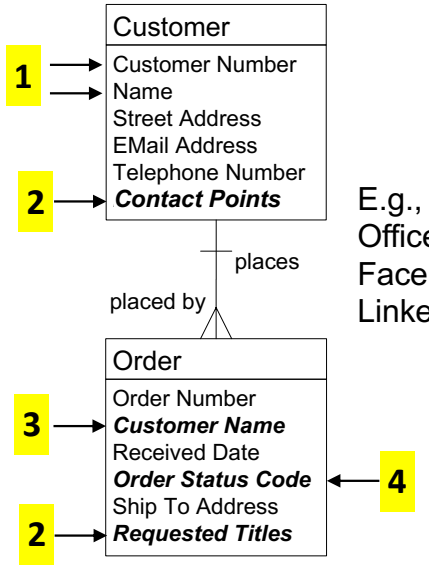
Have properties that we address during the transition from Concept Model to Logical Data Model

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, 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, …

Starting a data 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" includes artifacts (forms, spreadsheets, reports…,) systems, mechanisms, job titles, organisational structures, work (processes, activities, steps…,) and anything else that isn't a basic thing or fact about a thing

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

Read this on your own for reference – starting a concept 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.

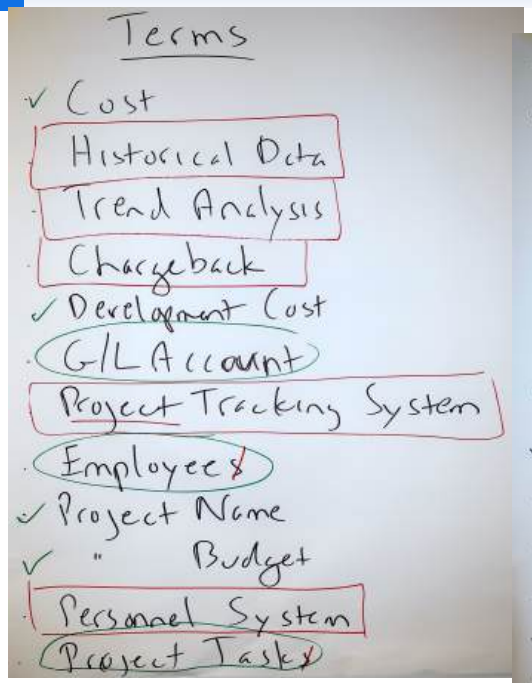
When a development project is initiated, a project name and a short description are recorded, among other things. Soon, before any further work is done on the project, a new account is created on the G/L System, identified by a G/L account number. Project costs will be charged to this account, and the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

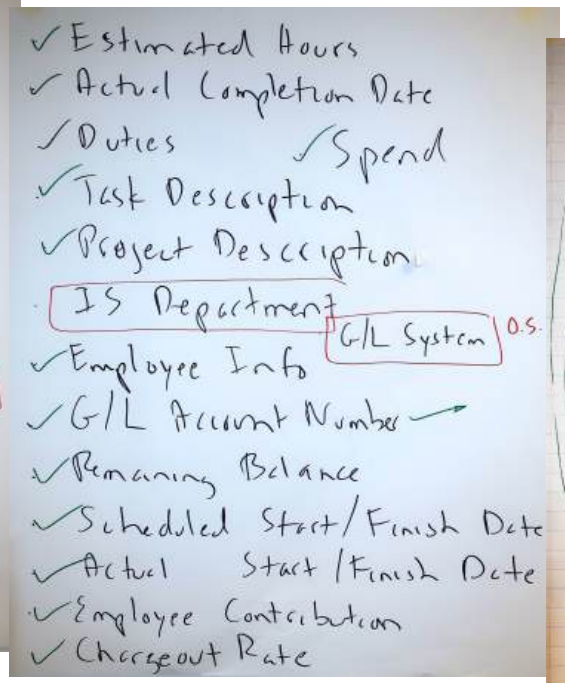
Project planners break a project down into many tasks, perhaps hundreds. A typical project task might be “Test Order Entry Module”. Some of the facts which are required about tasks include a brief task description, estimated work hours, and the scheduled start and finish dates.

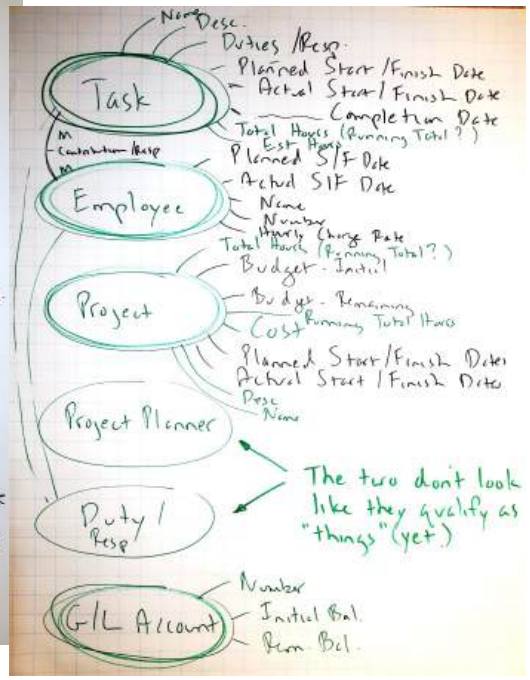
Eventually, individual employees are assigned responsibility for the tasks. Some tasks will be the responsibility of many employees, and an employee might be assigned to many tasks. As each employee is assigned to a project task, their planned start and finish dates, their contribution to the task (not 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 thing – *"what"* – with no reference to *"who or how."* Artifacts typically contain attributes from *multiple* Entities
   e.g., *"Admission Request Form"* or *"Orders Summary Spreadsheet"* or *"Daily Call Log"* or *"Class Roster" or "Materials List Fax"* or…

2. 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?*

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

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…?"

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.)

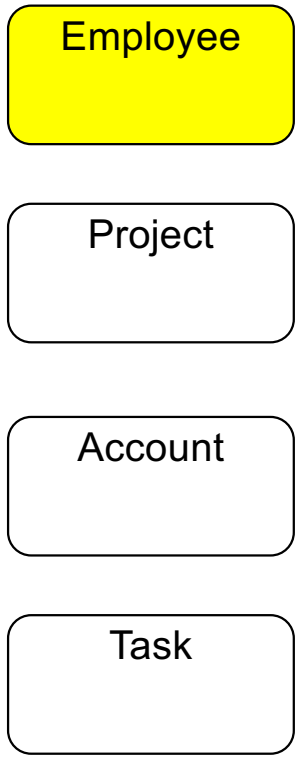
Entity definition format:

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

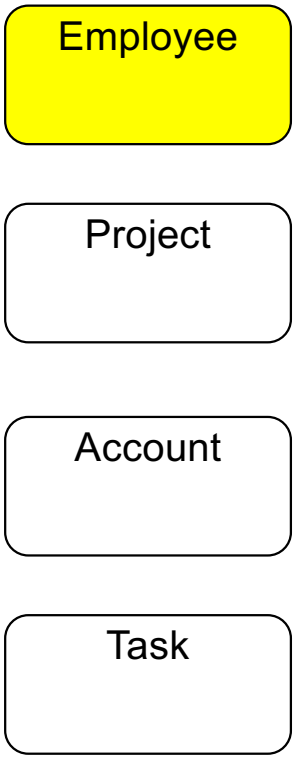
"Can anyone think of examples that might surprise someone else – that is, anomalies or potential sources of confusion."
E.g., how could we legitimately have different ideas what "Employee" means?

-
-
-
-
-
-
- ...



Brainstorming space

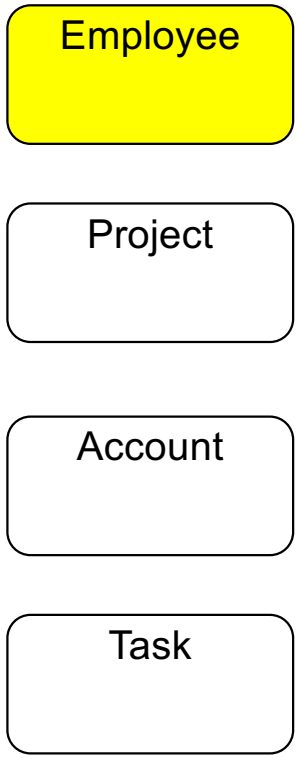
"Can anyone think of examples that might surprise someone else – that is, anomalies or potential sources of confusion."
E.g., how could we legitimately have different ideas what "Employee" means?



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)

For reference – 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

For reference – 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

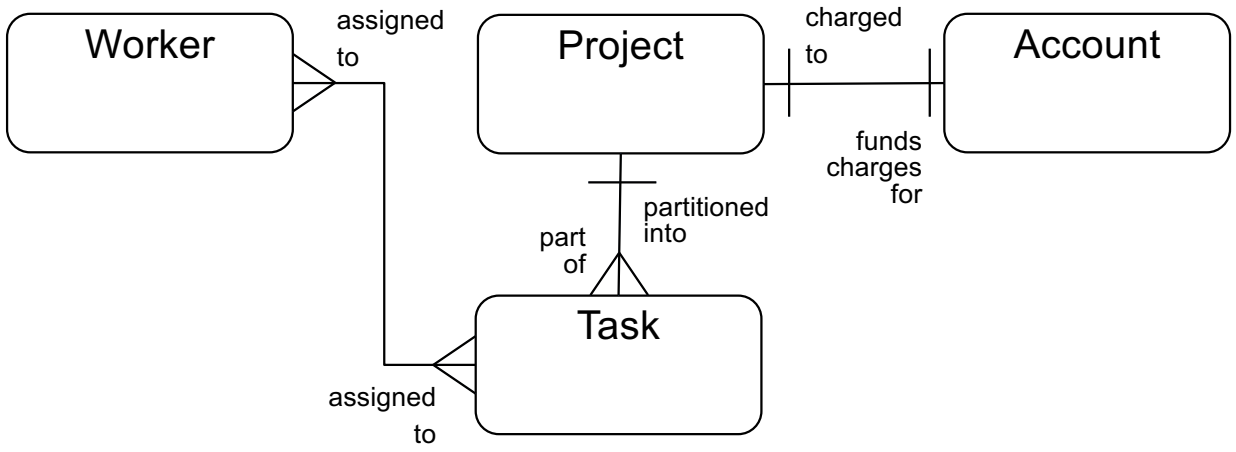
Worker

Project

Account

Task

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.)

Overview of topics

A Business Analysis framework

Concept Modelling as a Foundation

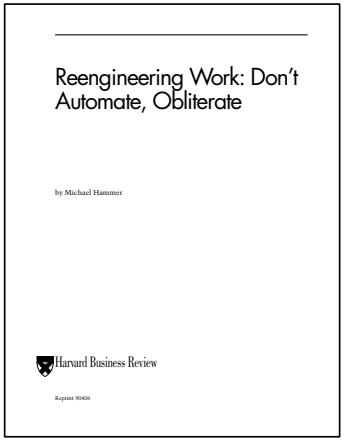
Business Process Essentials

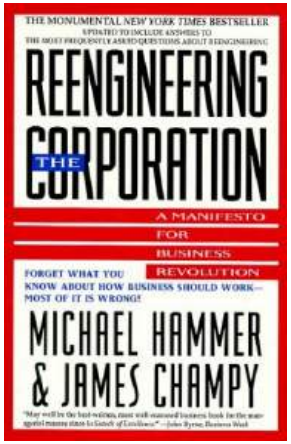
The Data-Process Connection

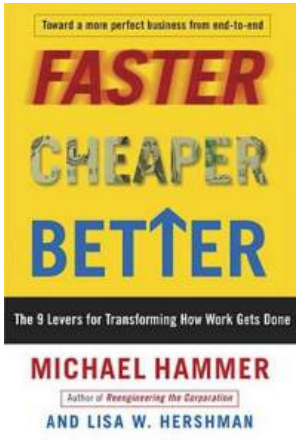
(Optional) A little more on Use Cases & Services

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

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.

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

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

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

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

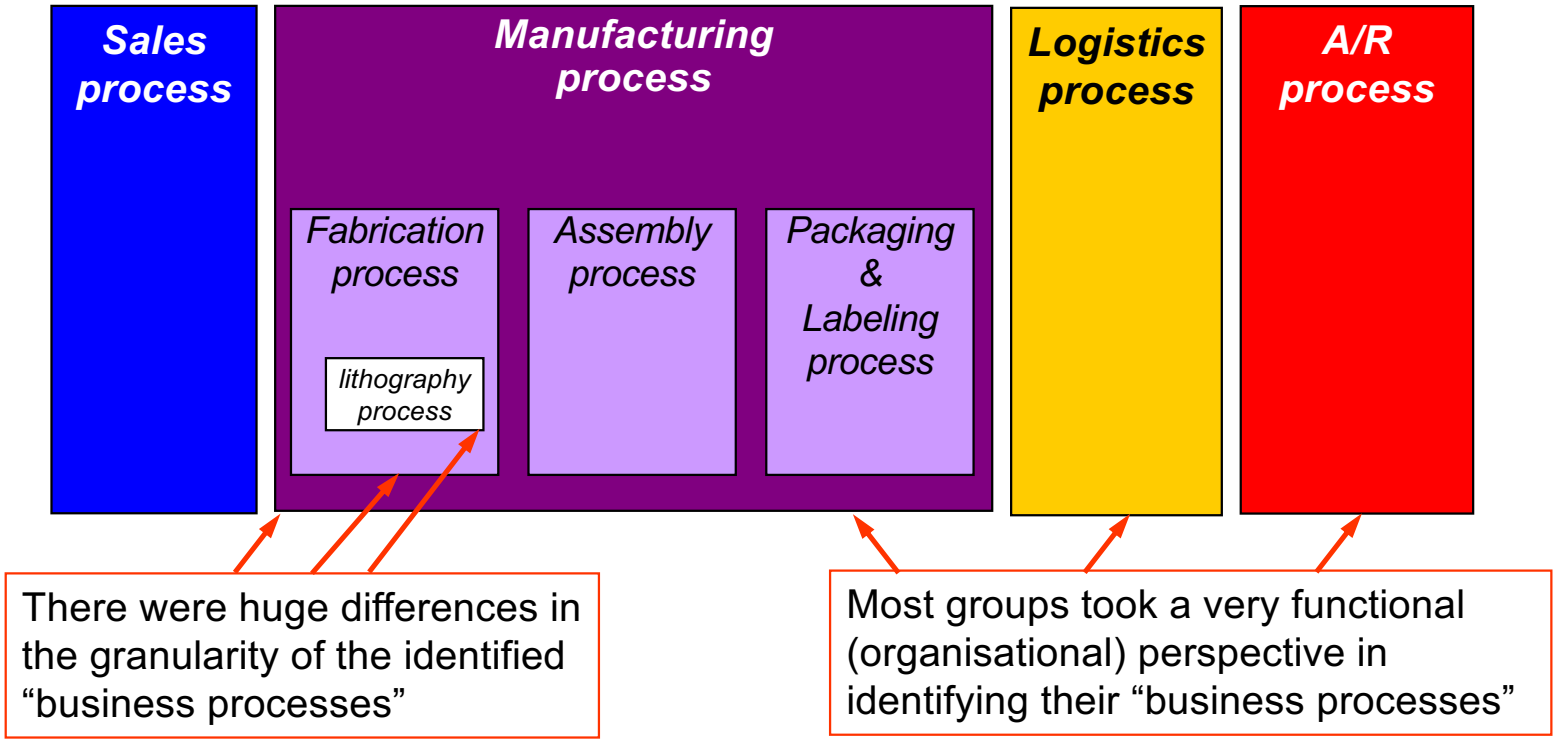
Seek balance – a "business process" lies between the extremes

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

The key issues – granularity and orientation

A real life (and expensive!) example

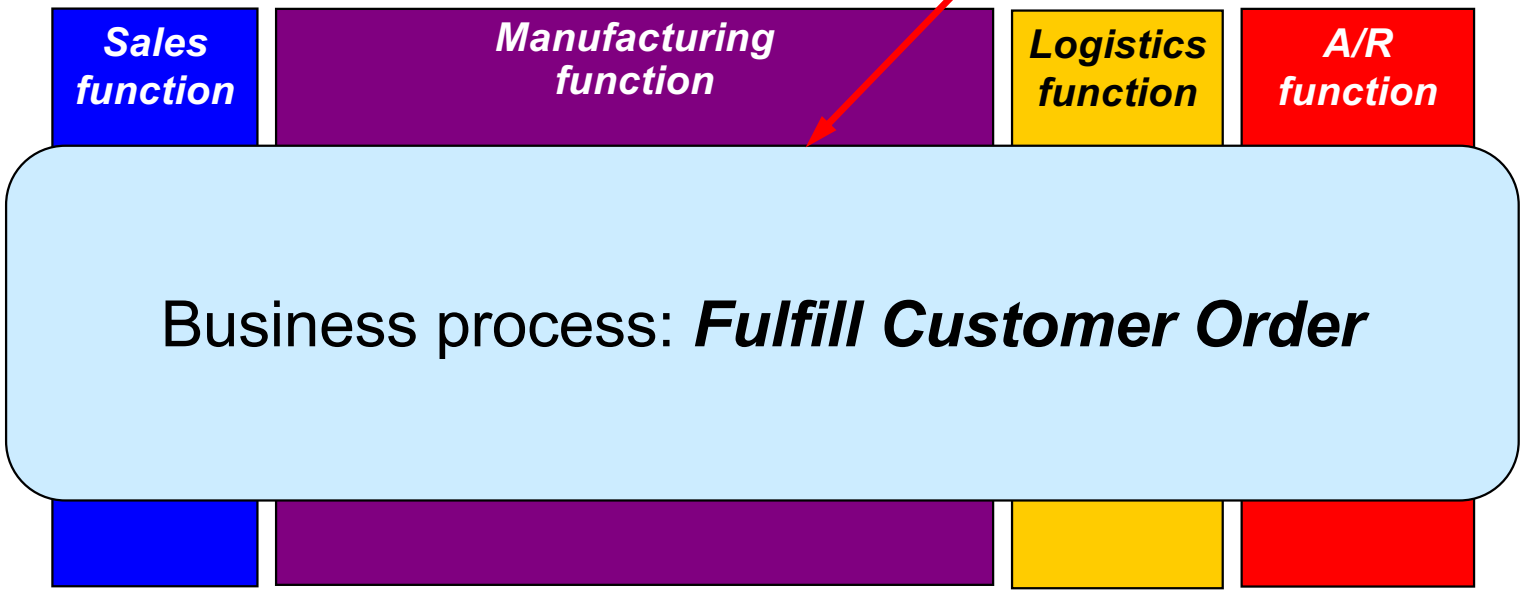
As part of a massive system implementation, a global manufacturer identified the *business processes* that were expected to improve:



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

The "real" business processes were missed

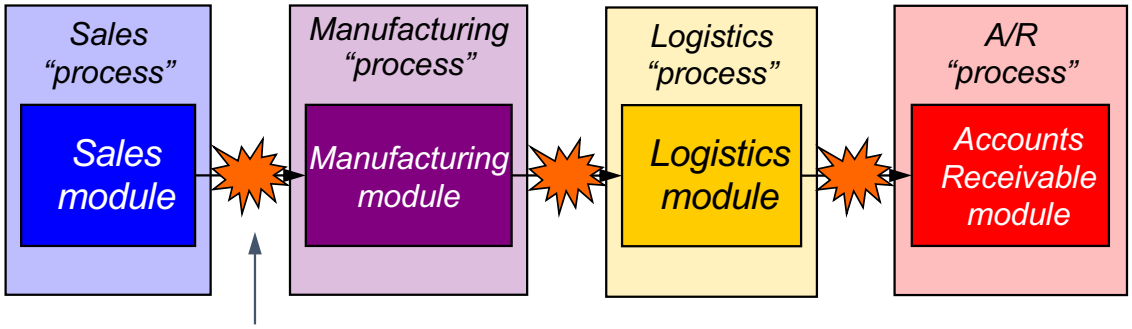
Everyone confused "process" and "function." None of the actual end-to-end processes were correctly identified.



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

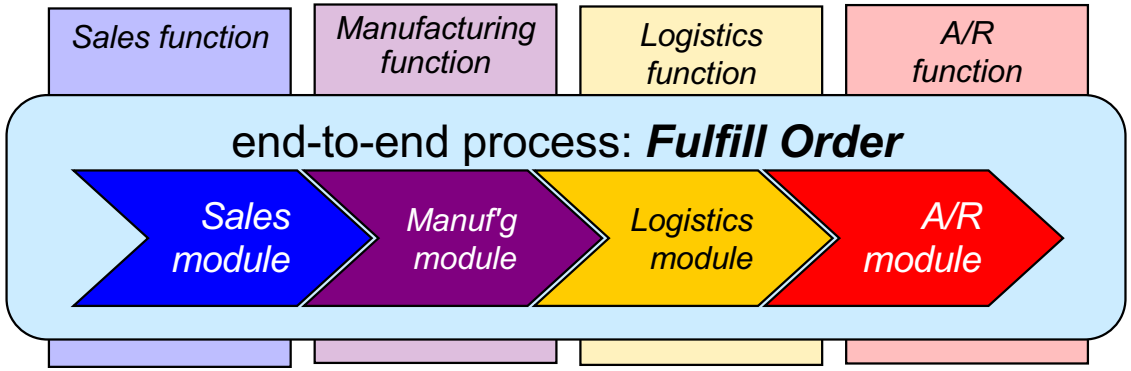
Impact of confusing function and process

Implementing SAP without clarity on "process":



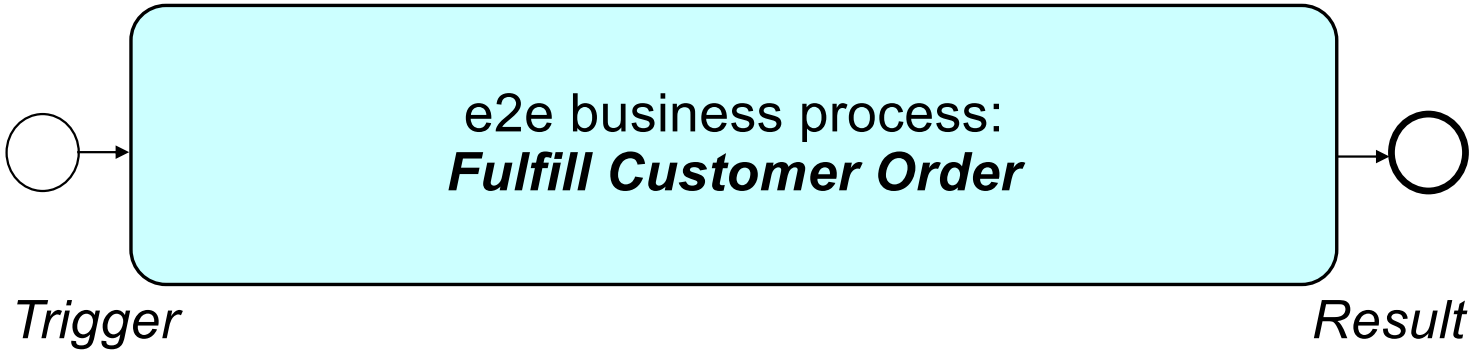
Conflicts: timing, coding, terminology, data formats, *performance targets*, …

SAP re-implemented in a process-driven configuration:

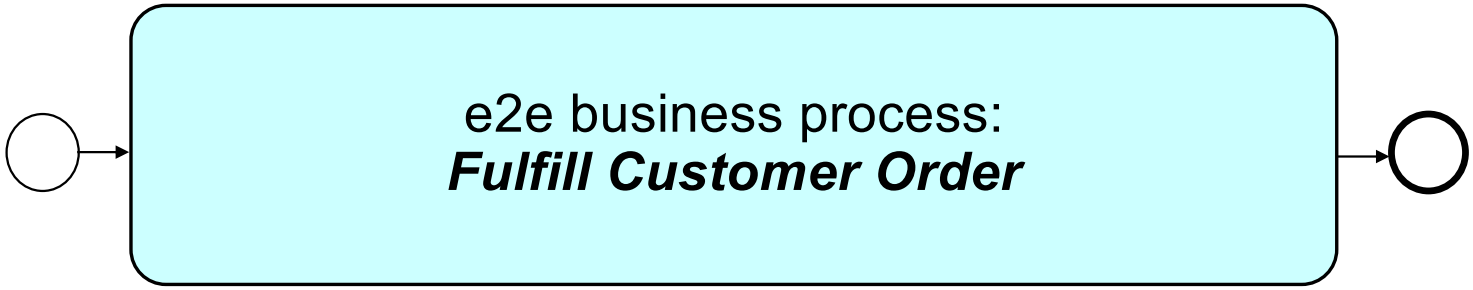


Same software, radically different outcomes

Discuss - what are the boundaries of the process?



What are the boundaries of the process?



Trigger

Order received? *No.*

Before that…

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

And before that…

- Demand is Signalled. *Yes.*

Result

Order is Shipped? *No.*

Order is Received? *No.*

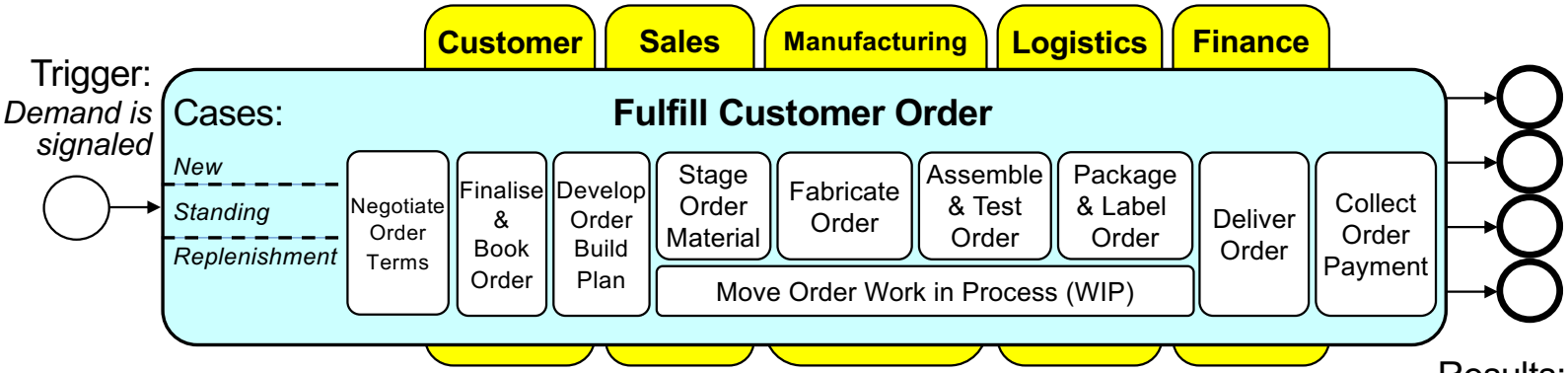
Order is Received, Tested, and Accepted? *Yes.*

Any other results?
Yes, for other stakeholders.

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

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

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



"TRAC" –

*1 – **T**riggering event or events*

*2 – **R**esults: final outputs*

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

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

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

*4 – **C**ases*

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

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

5 – Functions or Organisation Units

6 – Actors and responsibilities

7 – Systems, data sources, other mechanisms

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

Results:

Customer:
Goods received, tested, & accepted

Owner:
Payment received

Performer:
Commission credited

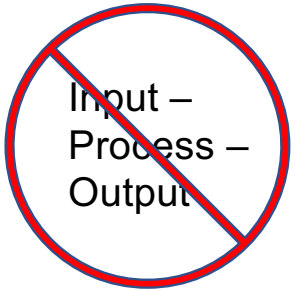
Industry Association:
Order stats reported

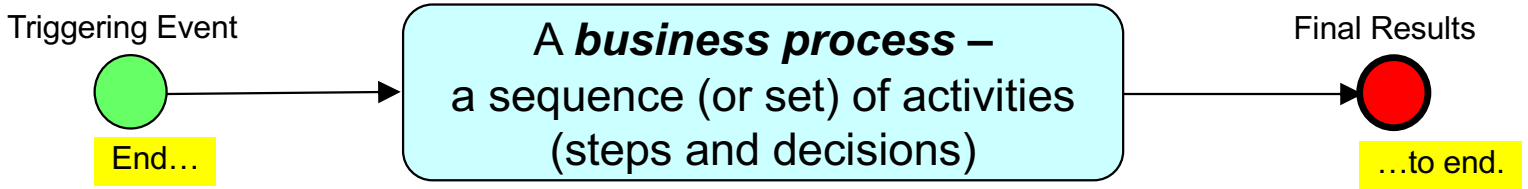
Always construct a *Process Scope Model* & a *Process Summary Chart* before diving into Workflow Modelling / Swimlane Diagramming

The essential framework

Business Process:

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





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

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

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

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

Naming conventions will make life easier

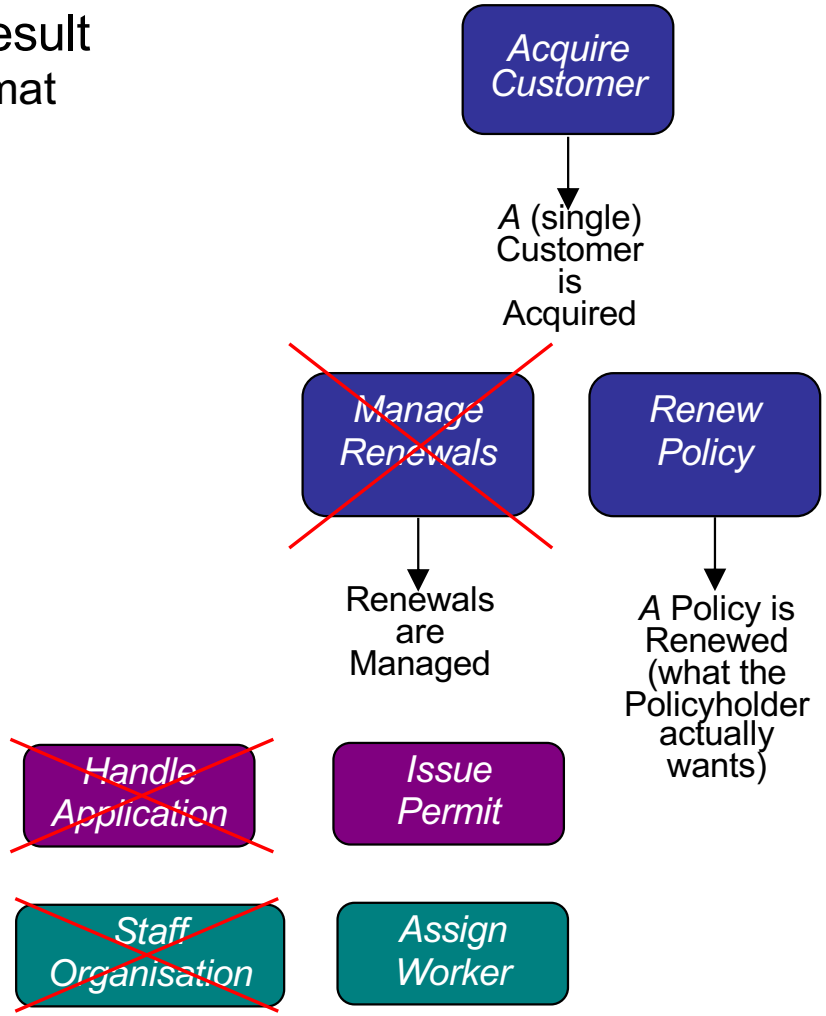
1. The process name ***must*** indicate the expected result
   - Name potential process in “active verb – noun” format
   - Restate that name as a result (“noun is verbed”)
   - Ensure this is the intended result of the process: *discrete*, so results are *identifiable* & *countable*

   - ***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, …

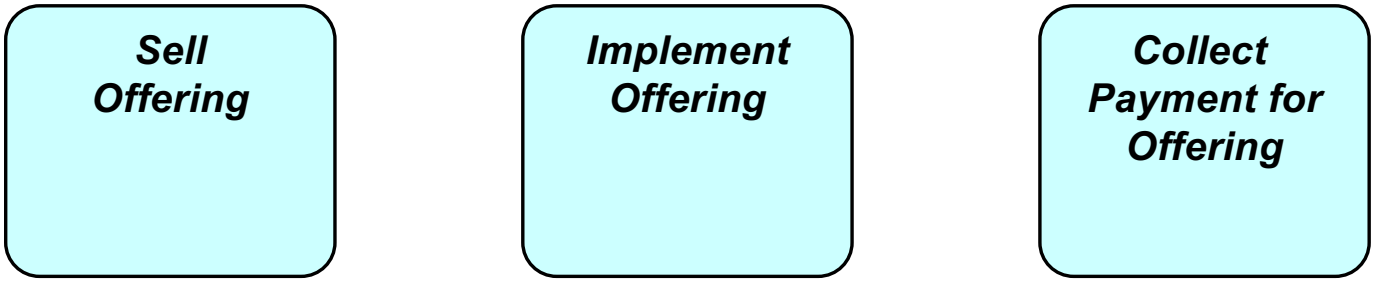
2. Name process from customer's perspective (what do they want from the process?)

3. Name process in the singular

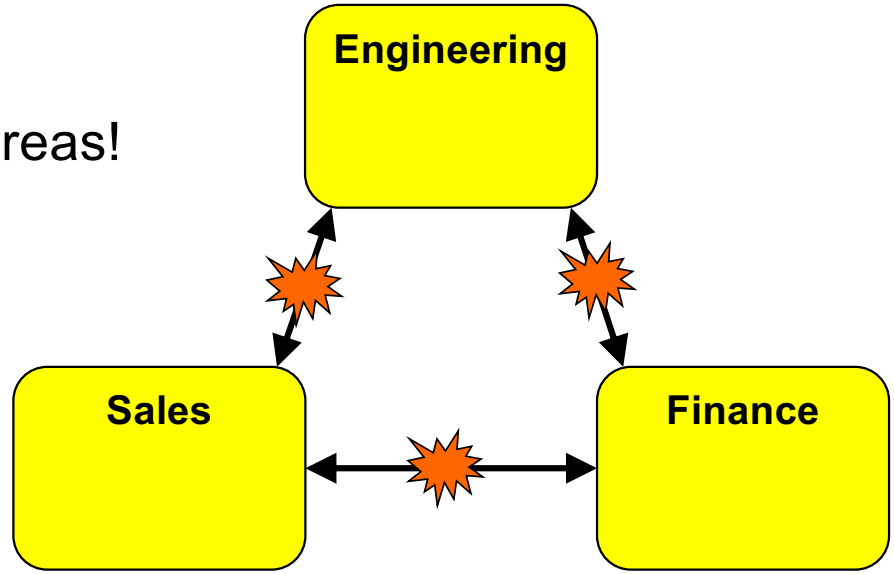


Another Business Process example, if we have time

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

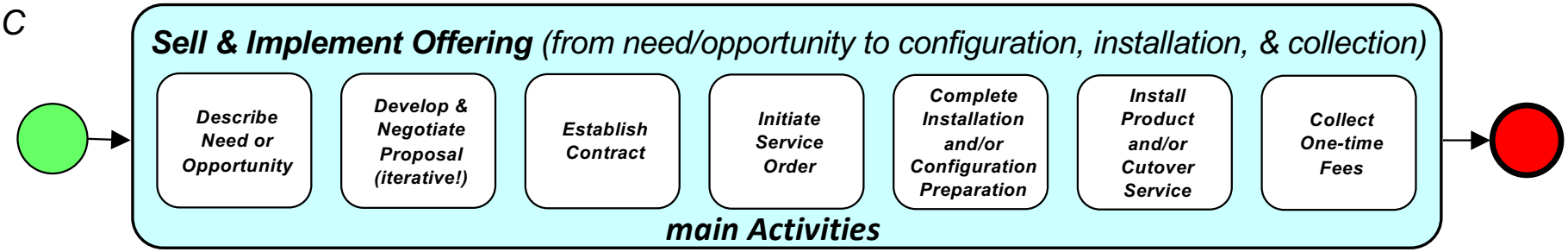


The outcome…
conflict between functional areas!



Process Scope Model showed ONE process not THREE

TRAC



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

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

Other factors:
- TBD

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

Employee:
- *Commission* or *referral credit*

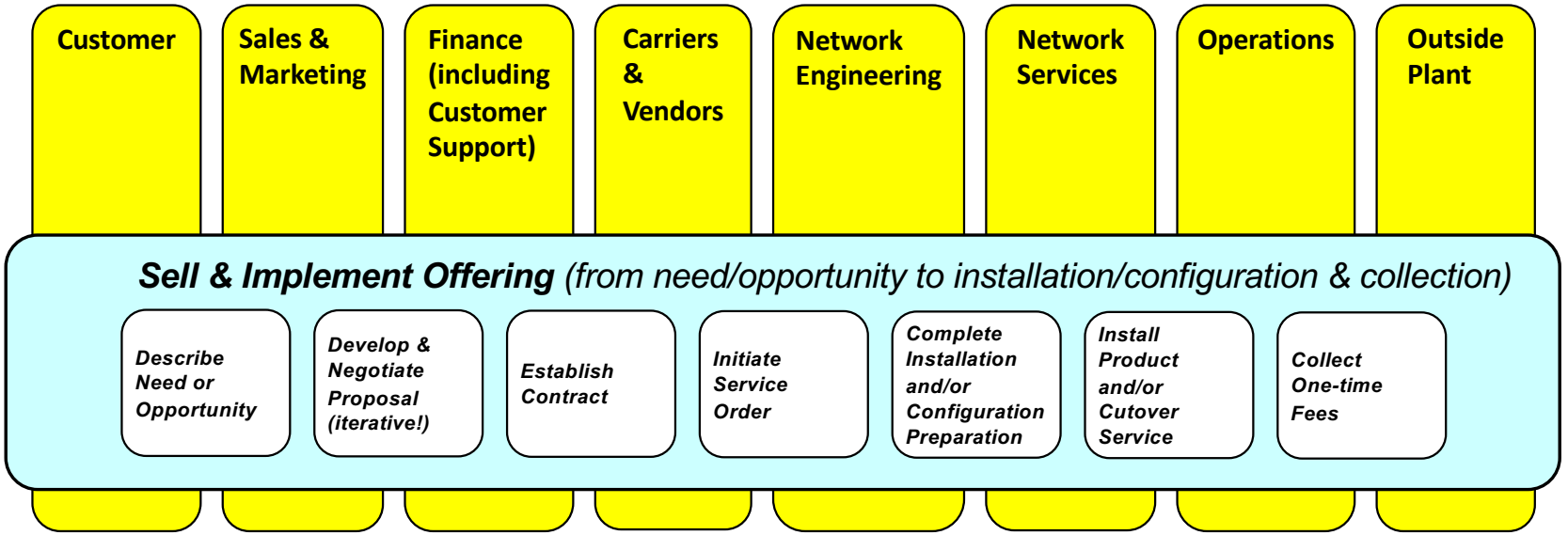
Agent:
- *Commission*

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

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

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

Process Summary Chart – my favourite diagram!



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

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

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

Multiple roles by organisation for "Sell & Implement Offering"

Customer

Roles:

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

Sales & Marketing

Roles:

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

Finance (including Customer Support)

Roles:

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

Carriers & Vendors

Roles:

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

Network Engineering

Roles:

- System Admins (assign IP)

Network Services

Roles:

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

Operations

Roles:

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

Outside Plant

Roles:

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

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

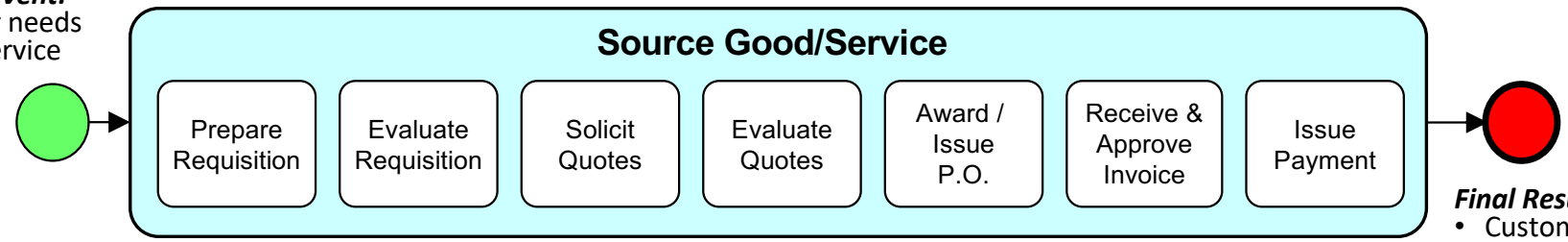
Another fast Augmented Scope Model example

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

Assume everything <$5000 is purchased with a PCard

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

Triggering Event:
- Customer needs Good / Service



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

Prepare Requisition

Develop scope of work / specs

Investigate potential vendors (and price?)

Solicit vendor quotes (just to get an idea)

Obtain approval (Department)

Verify Item and Account (General Accounting)

Submit requisition (visible to all)

Evaluate Requisition

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

Assign (or re-assign Buyer as necessary)

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

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

Solicit Quotes

Determine (additional) potential vendors

Solicit quote (including Bid Due Date)

Post quote (solicitation documents) in "the binder"

Resolve vendor queries

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

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

Evaluate Quotes

Receive quote (mail, fax, e-mail, …)

Confirm completeness

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

Clarify (not negotiate) with vendor

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

Identify vendor

Award / Issue P.O.

Generate Purchase Order

Notify Requestor

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

Receive & Approve Invoice

Receive Good/Service
* Invoice could be attached

Accept Good/Service

Issue invoice (vendor)

Issue Payment

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

* Vendor complains invoice is "lost"

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

If <$5000, match
- invoice
- PO

* Could invoice $4K on $40K PO

Batch invoices for GAD

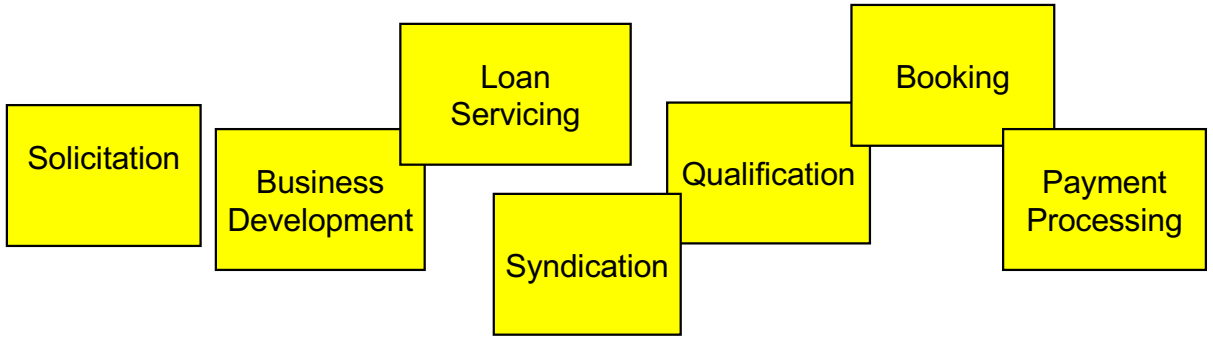
Issue Payment (Magic Happens Here)

Receive payment

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

Process discovery example

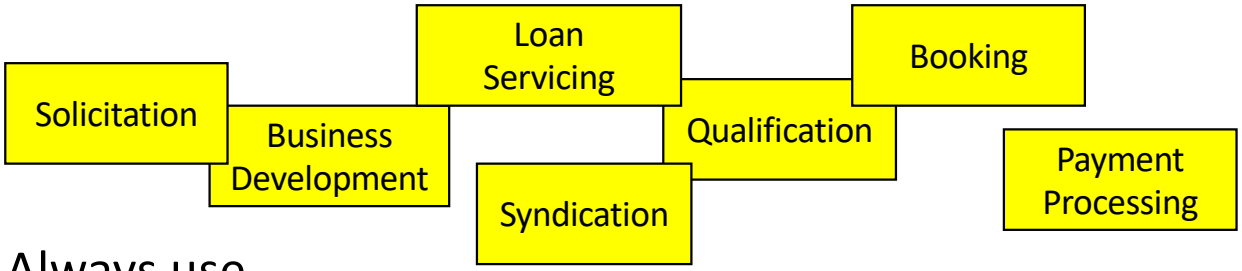
A bank believed they had identified the 12 *business processes* in their Commercial Loans Management area, including these 7:



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



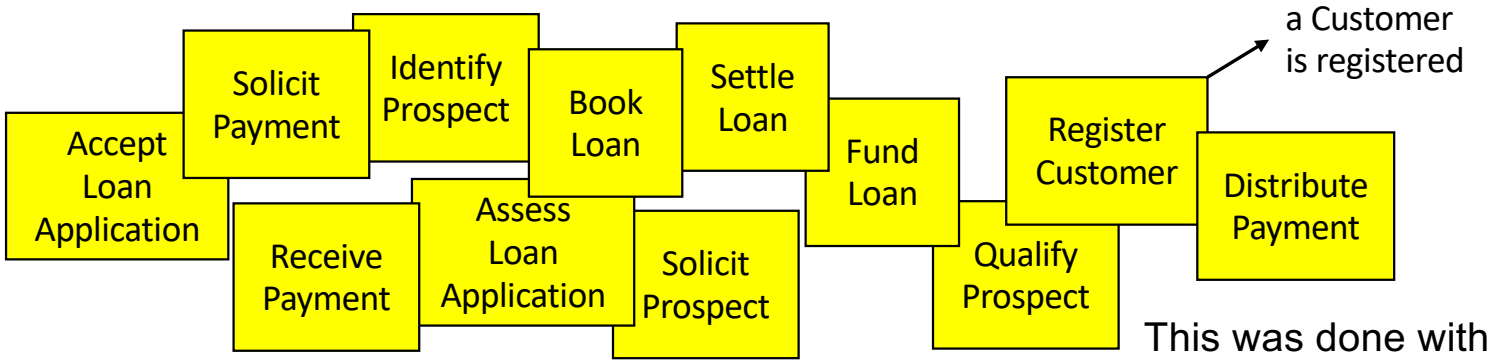
Dubious "business processes"

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

~~Solicitation???~~ Solicit… *what?* Solicit Prospect Solicit Loan Payment

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

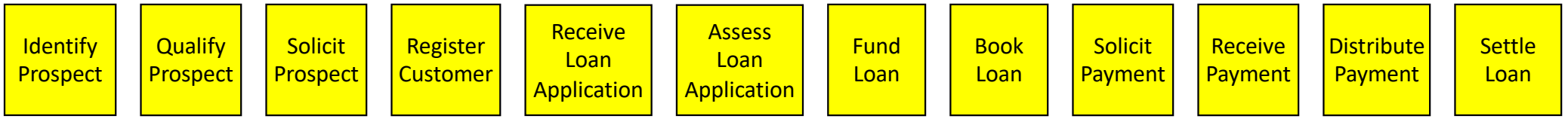
Can you put these in sequence?



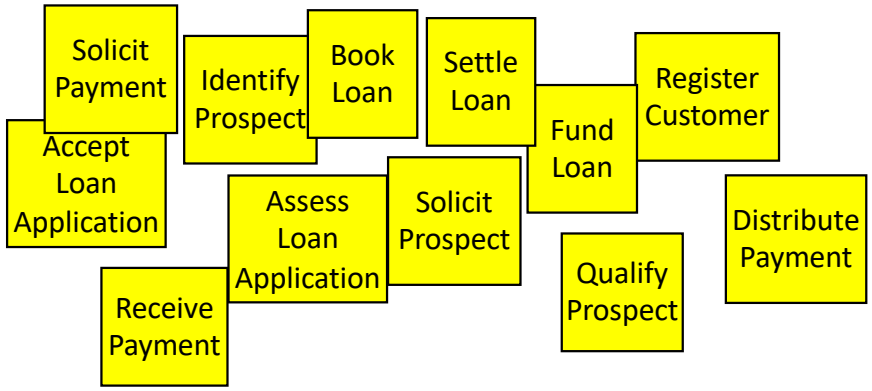
This was done with Post-its and flipcharts…

Sequence activities

Not usually linear – parallel chains are typical



Have the clients arrange the activities in sequence:
- *easy!*
- *a learning experience!*

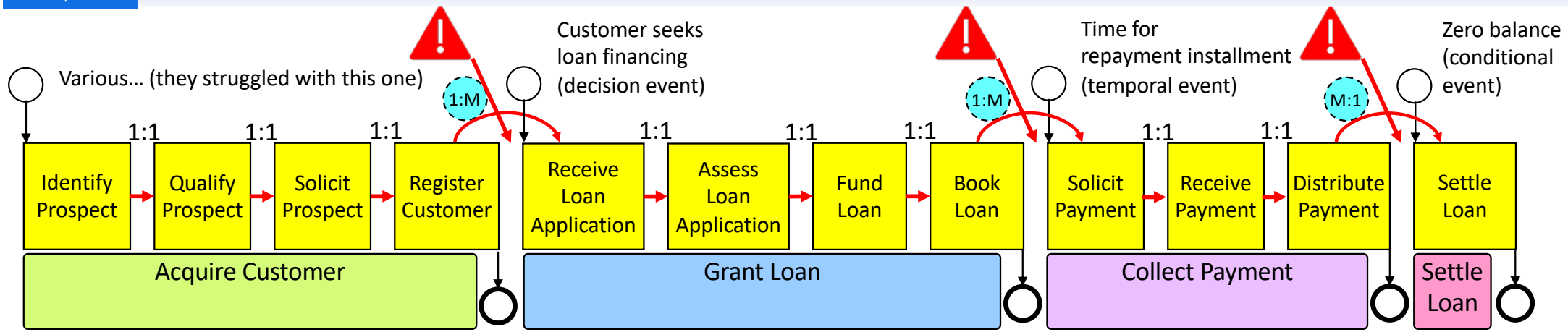


Now we'll use my "TRAC" framework for business processes –

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

This was done with Post-its and flipcharts…

Summary – use TRAC to discover business process boundaries



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

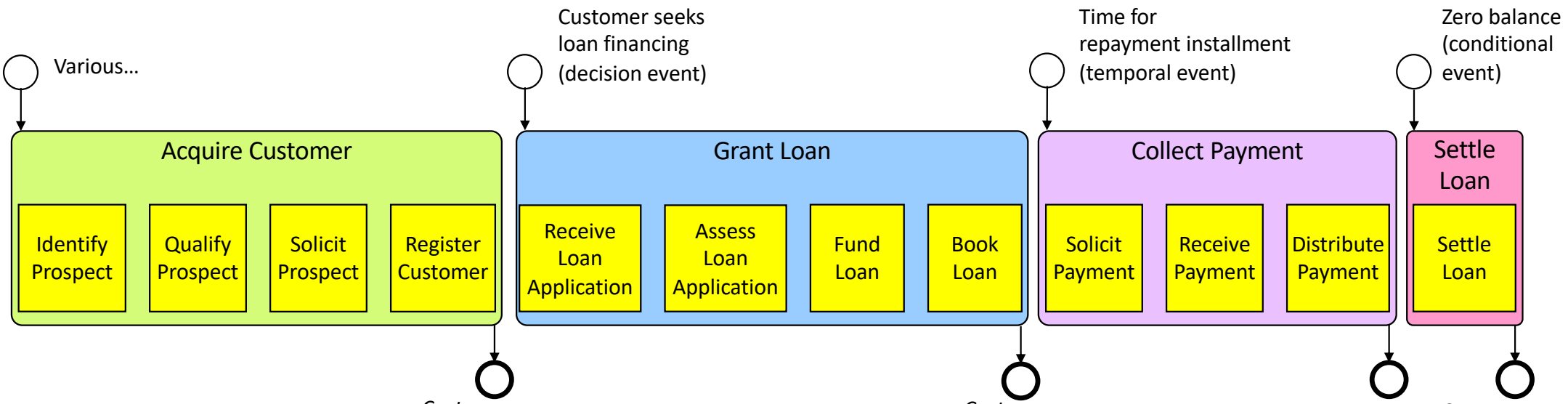
Customer:
Loan funds available
The Bank:
a performing asset (Loan)
Syndication Partners:
a share of the Loan

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 beyond the organisation’s control is required before activities can proceed (decision, time, condition)
3. Identify “cardinality” of connections between Activities (1:1, 1:M, M:1)
4. Identify “tokens” flowing through the activities

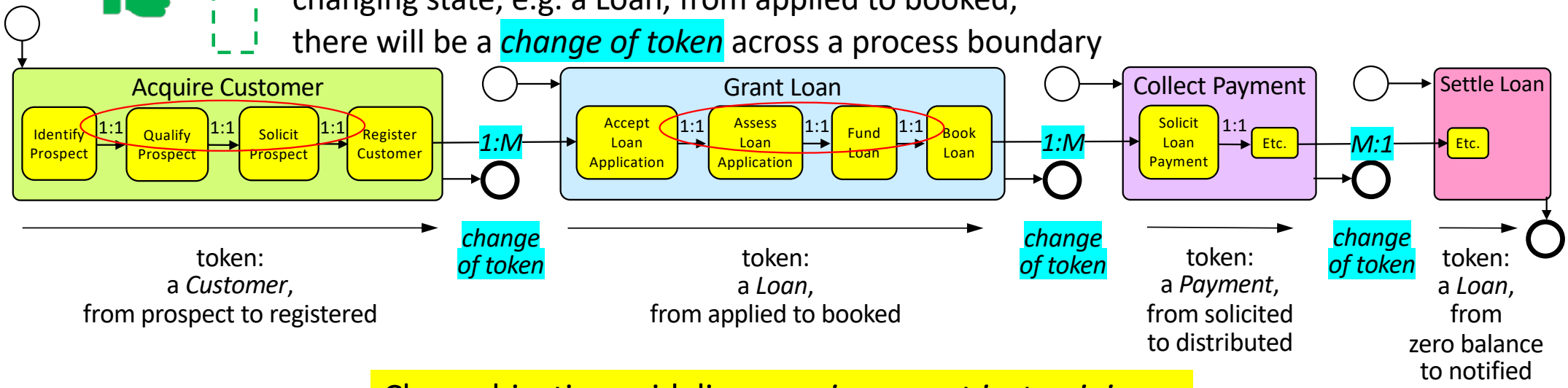
Four end-to-end business processes, objectively demonstrated



Client had faith these were *their* business processes

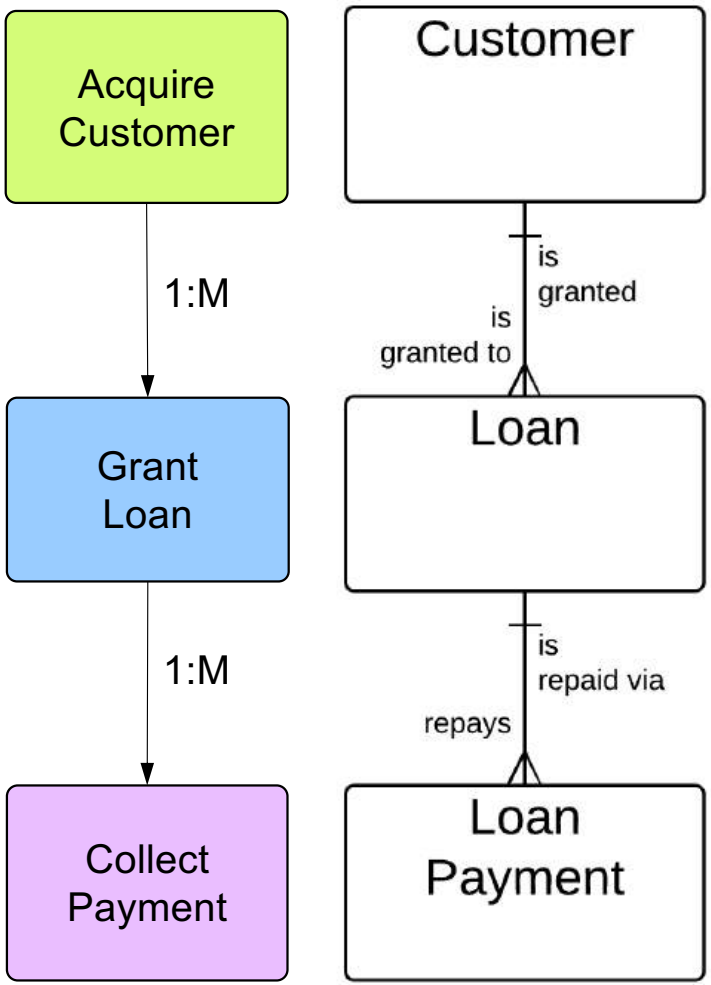
Six guidelines for well-formed processes

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*

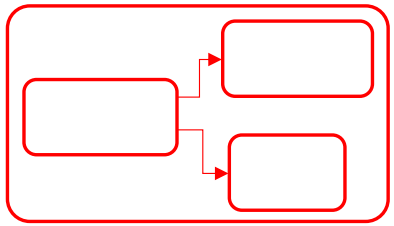
Correspondence to the Business Object Model



- The nouns in your verb-noun *Process* name are most often the *Business Objects* in your Business Object 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 Business Object
  - Assess *Customer* Performance
  - Retire *Customer*
  - Merge *Loans*
  - Write Off *Loan*
  - …

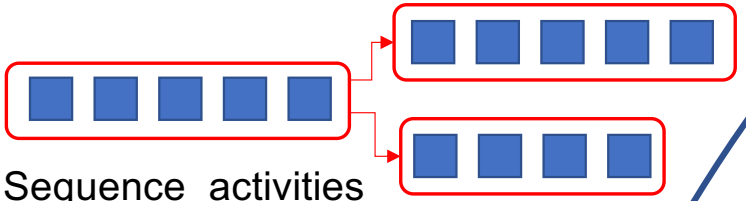
The arc of modelling and analysis

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

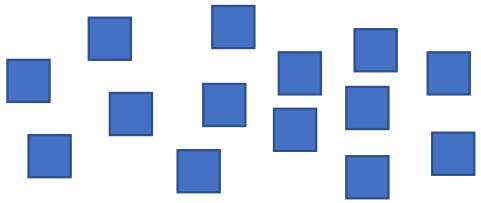


Draw a *Process Landscape,* our overall framework

Use objective criteria to identify process boundaries



Sequence activities into "chains"



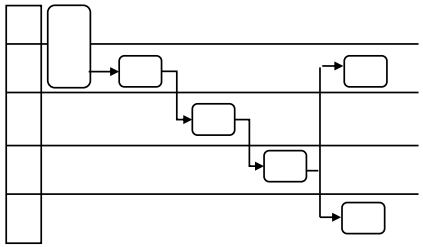
Granular activities (real, recognisable work)

bottom-up synthesis

top-down analysis

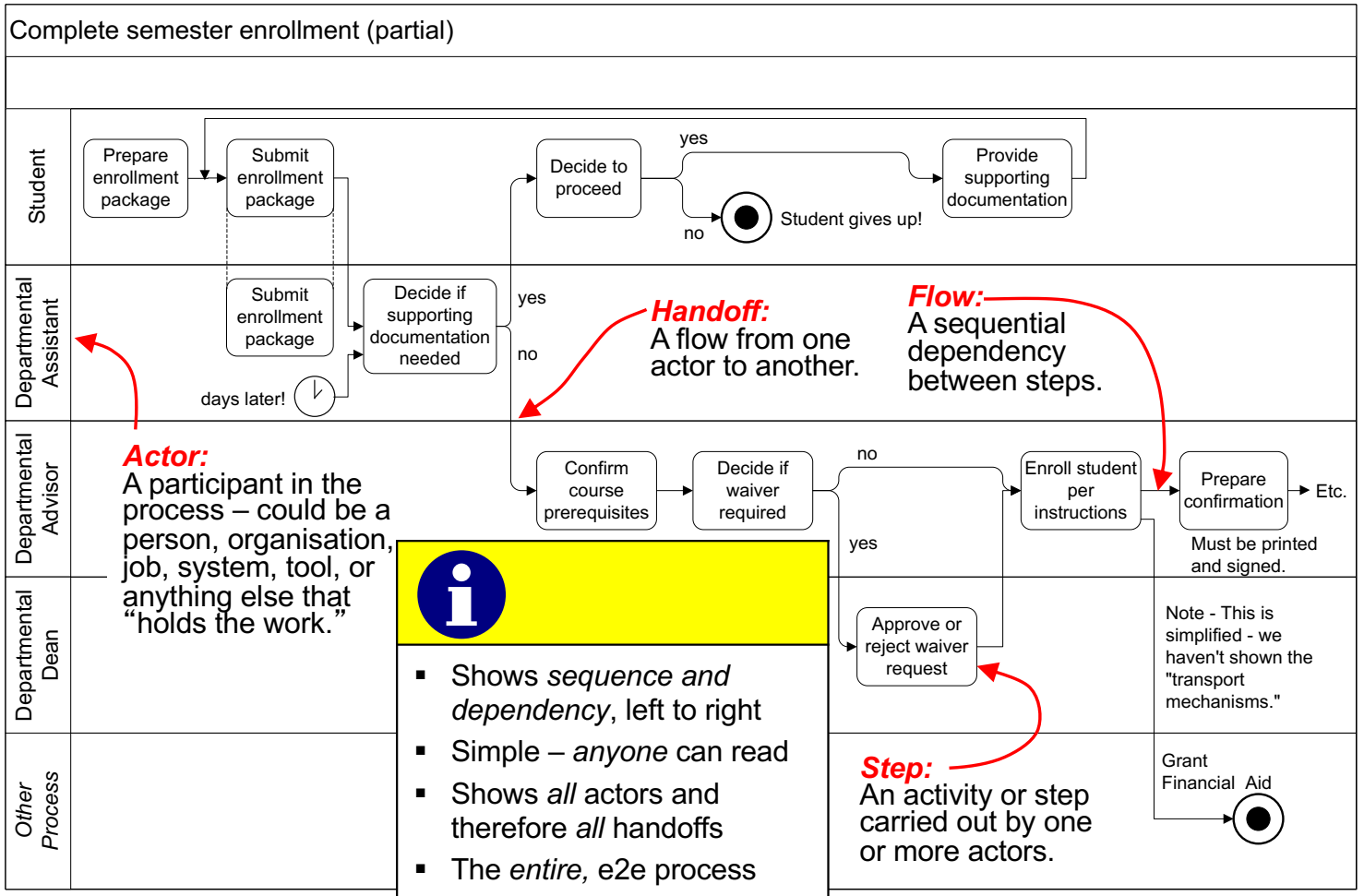
Progressive refinement and elaboration

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



Develop *Initial Swimlane Diagram*

Simple Swimlane Diagrams – maximise their strengths



Who – the actors

What – the steps

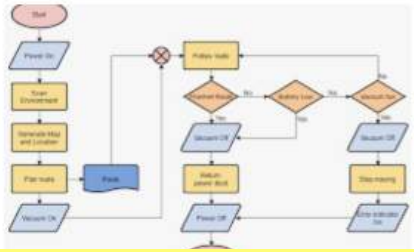
When – the flow

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

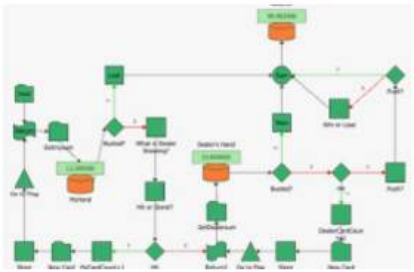
- step-by-step procedures
- checklists
- decision trees
- use cases
- etc.

Why did simple Swimlane Diagrams become popular?

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



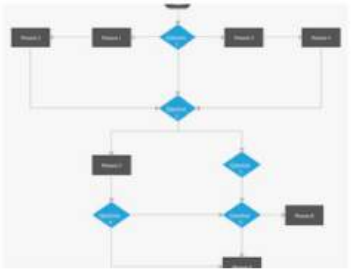
Dota 2 Flow Chart ...
reddit.com



complex RENO flowcharts easier ...
weibull.com

… lots of diagrams I might draw differently.

Follow flowchart best practices without ...
cacoo.com

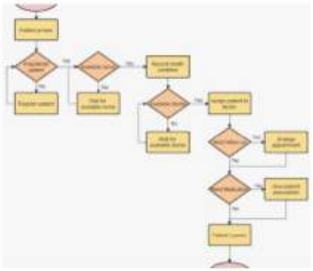


Flowchart Tutorial (Complete Flowchar...
creately.com

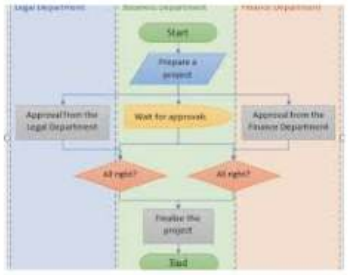
21 Creative Flowchart ...
visme.co

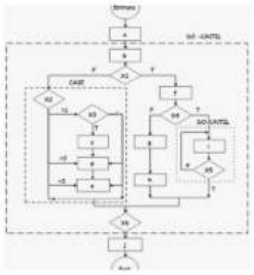
Flowchart Programming ...
conceptdraw.com

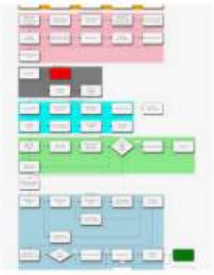
Free Flowchart Templates ...
gliffy.com

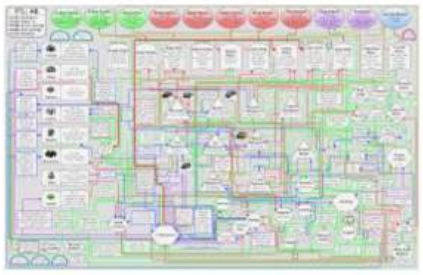


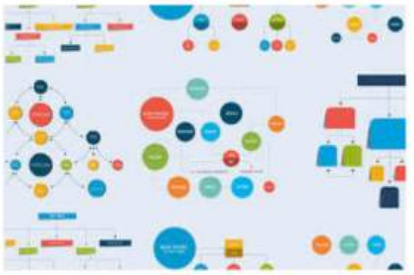
Flowchart Tutorial (with Symbols, ...
visual-paradigm.com

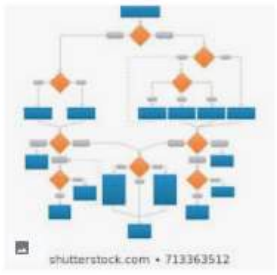




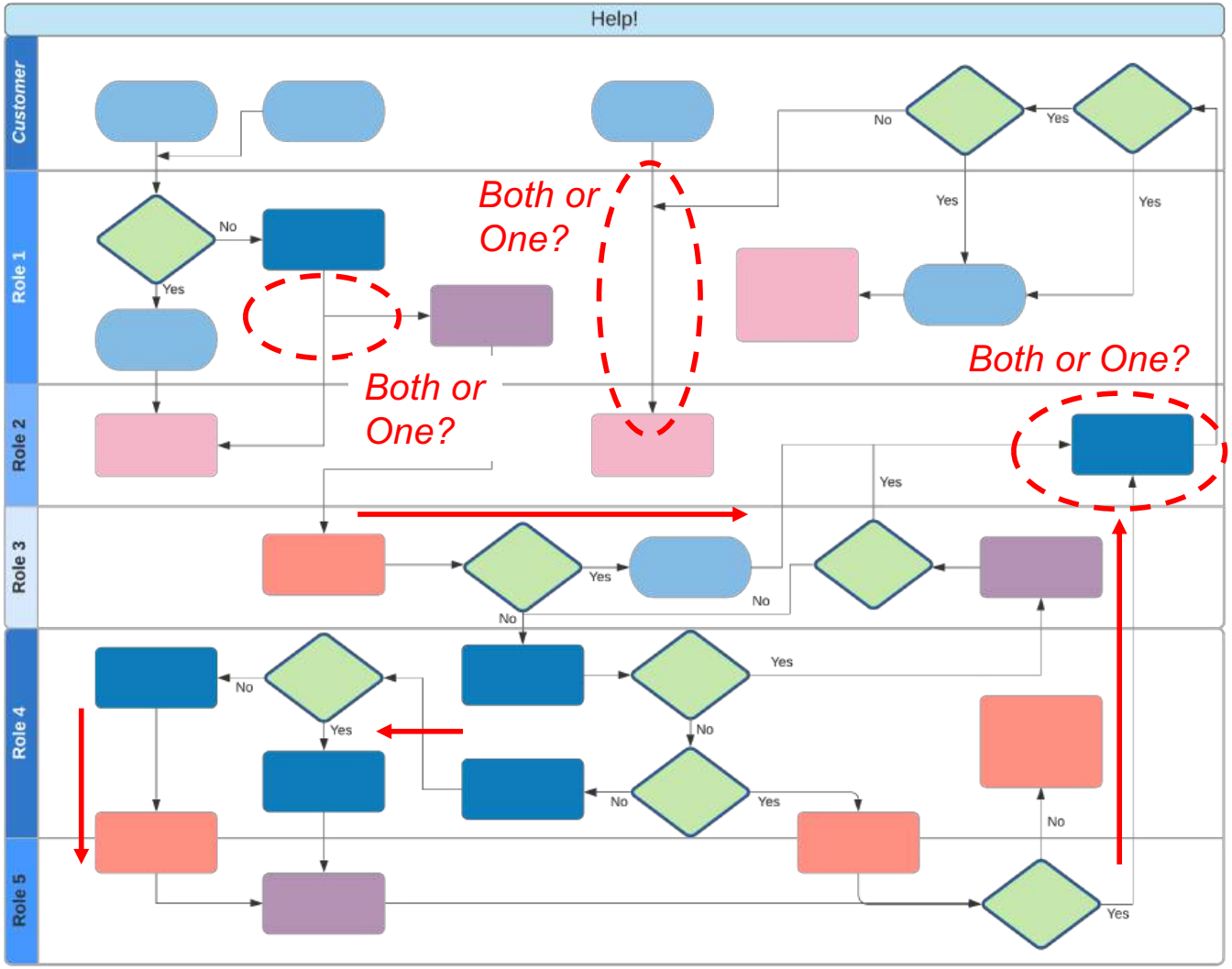








One example… "Chaos With Colours"



Probably accurate, and not too many symbols, but…

- are different colours helpful?
- 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.

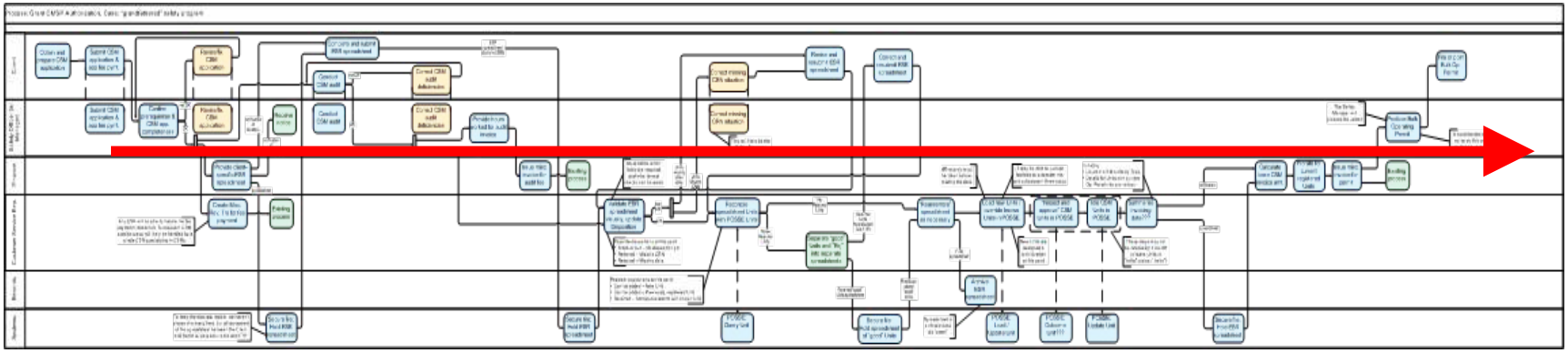
Core principles – "Flow first, detail later" and "Simplicity!"

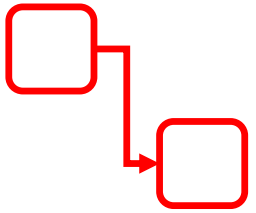
The purpose of a *Workflow* Model is to show the *Flow* of *Work*

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

Overview of topics

A Business Analysis framework

Concept Modelling as a Foundation

Business Process Essentials

The Data-Process Connection

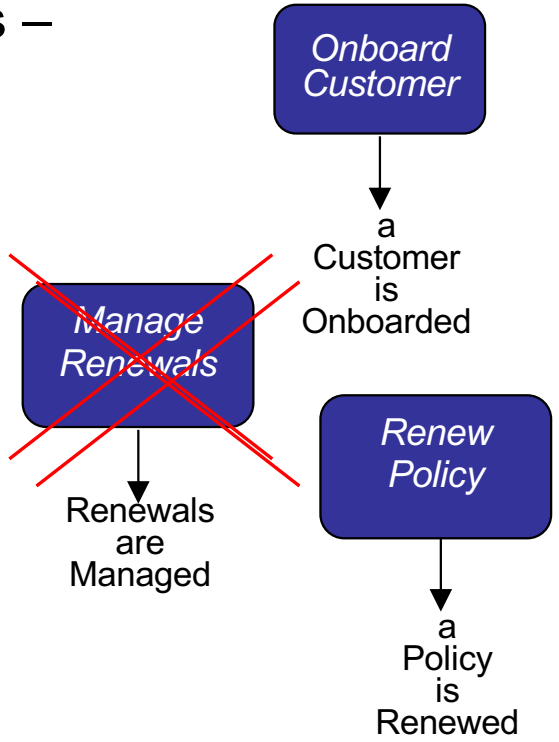
(Optional) A little more on Use Cases & Services

Process and Data people often miss the obvious connection!

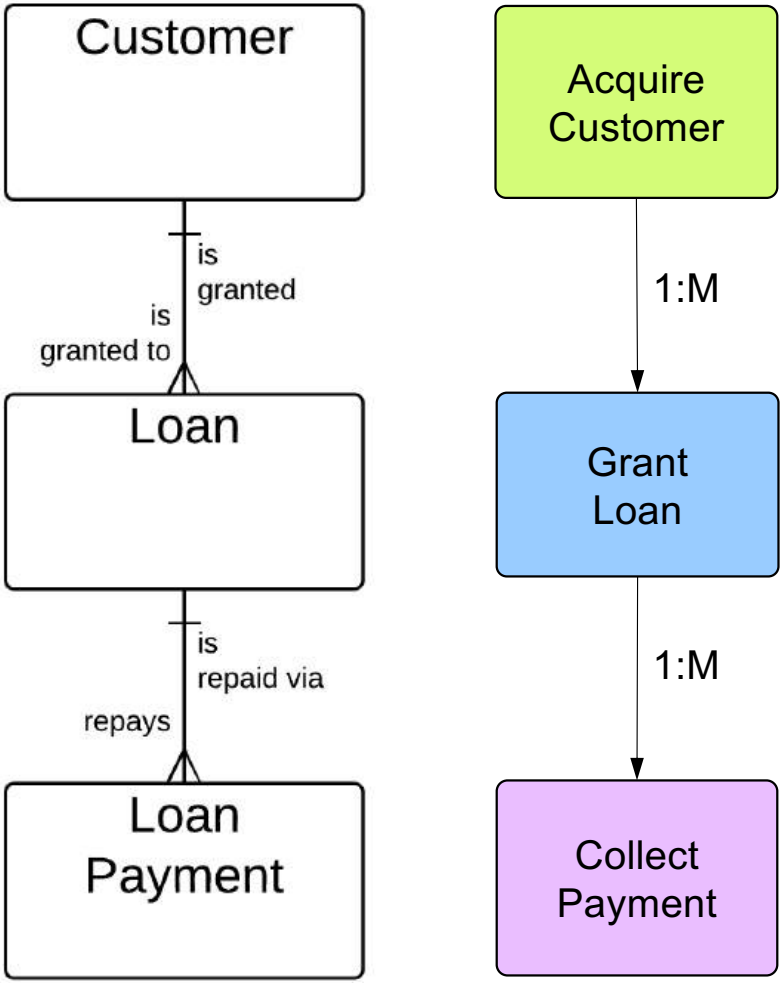
First, a naming convention for Business Processes helps – a good process name ***must*** indicate the expected result:

- Name process in "verb – noun" format
- Restate that name as a result – "noun is verbed"
- Is this is the intended result of the process?
  Is it a *discrete* result, so results are *identifiable* & *countable?*

The *noun* in the verb – noun pair is most often an *entity* from the concept model



We showed this connection earlier



The nouns in your verb-noun *Process* name are most often the *Entities* in your Concept Model;
each will usually have one primary *Process*

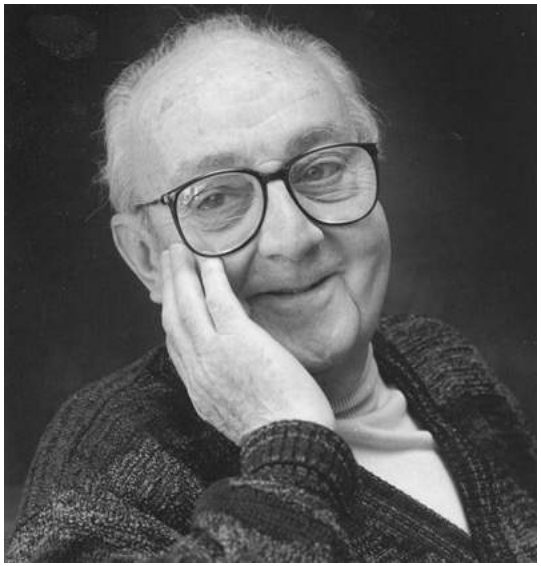
The relative number of Process instances (e.g., 1:M or M:1) align with relationship cardinality

This *does not* mean there is only one Process per Entity

- Assess *Customer* Performance
- Retire *Customer*
- Merge *Loans*
- Write Off *Loan…*

A core idea – "essential" models

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



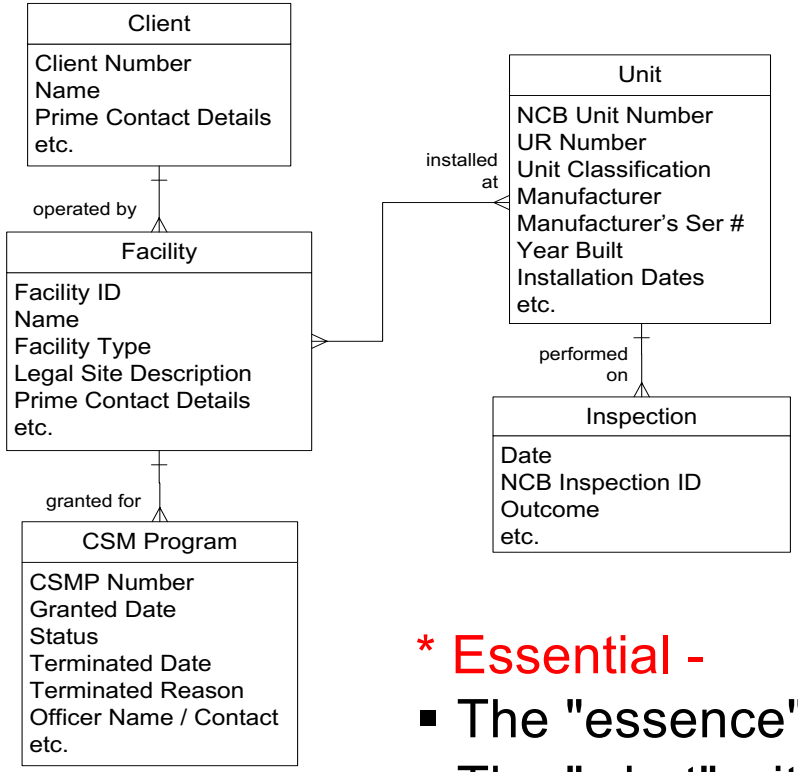
George E. P. Box
1919–2013

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.

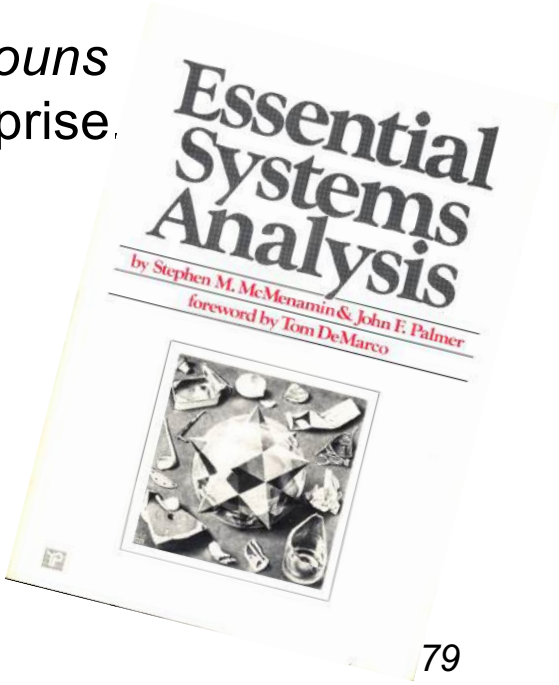
Concept Model – an Essential* model



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

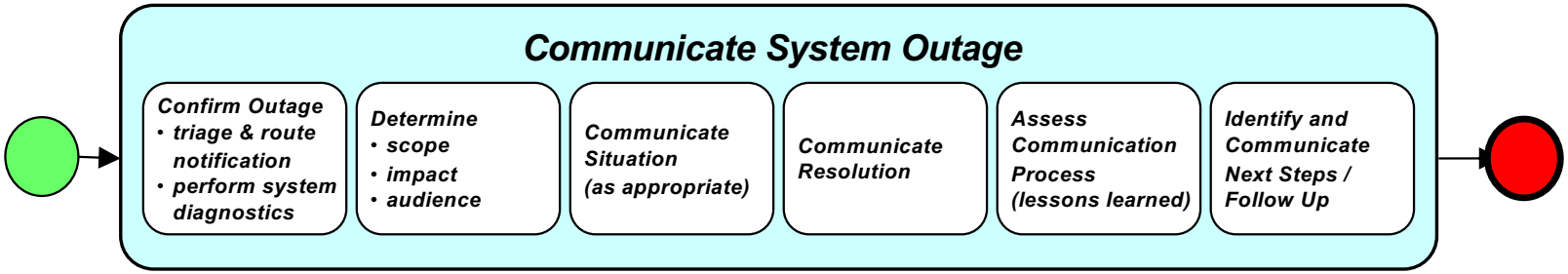
A *shared language* of the *nouns* that are central to the enterprise.
Always start here!



* Essential -

- The "essence" of the subject
- The "what" with no reference to "who" (role or organisation) or "how" (implementation or technology)

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:

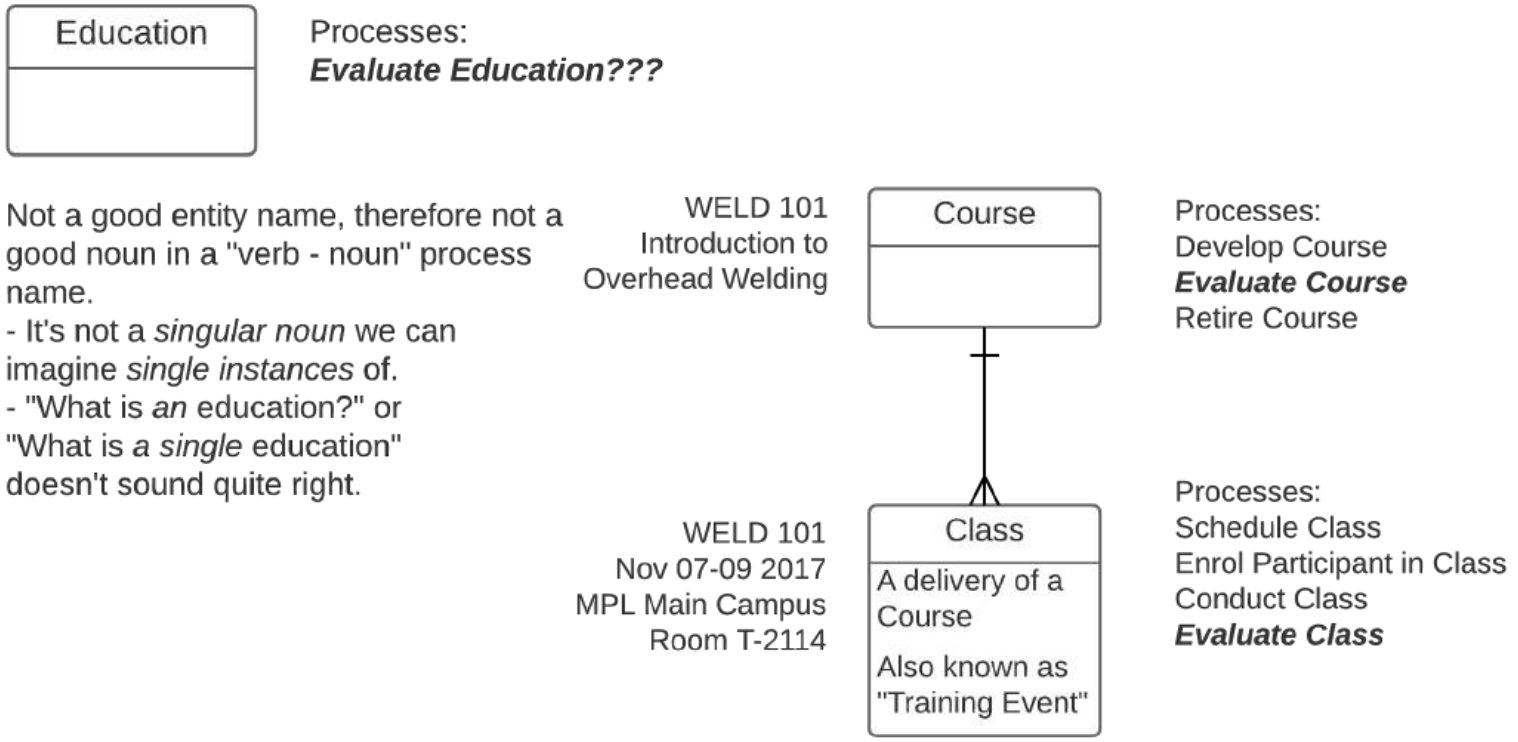
Service is restored and root cause is known (or is determined to be unknowable) and resolution is communicated:
- Externally ("good news")
- Internally ("cause & resolution)

Process Scope Model using "TRAC" -
what is the Trigger, *what* are the Results,
what are the main Activities
(5 to 7 milestones, phases, or subprocesses,)
and *what* are the main cases or variations?

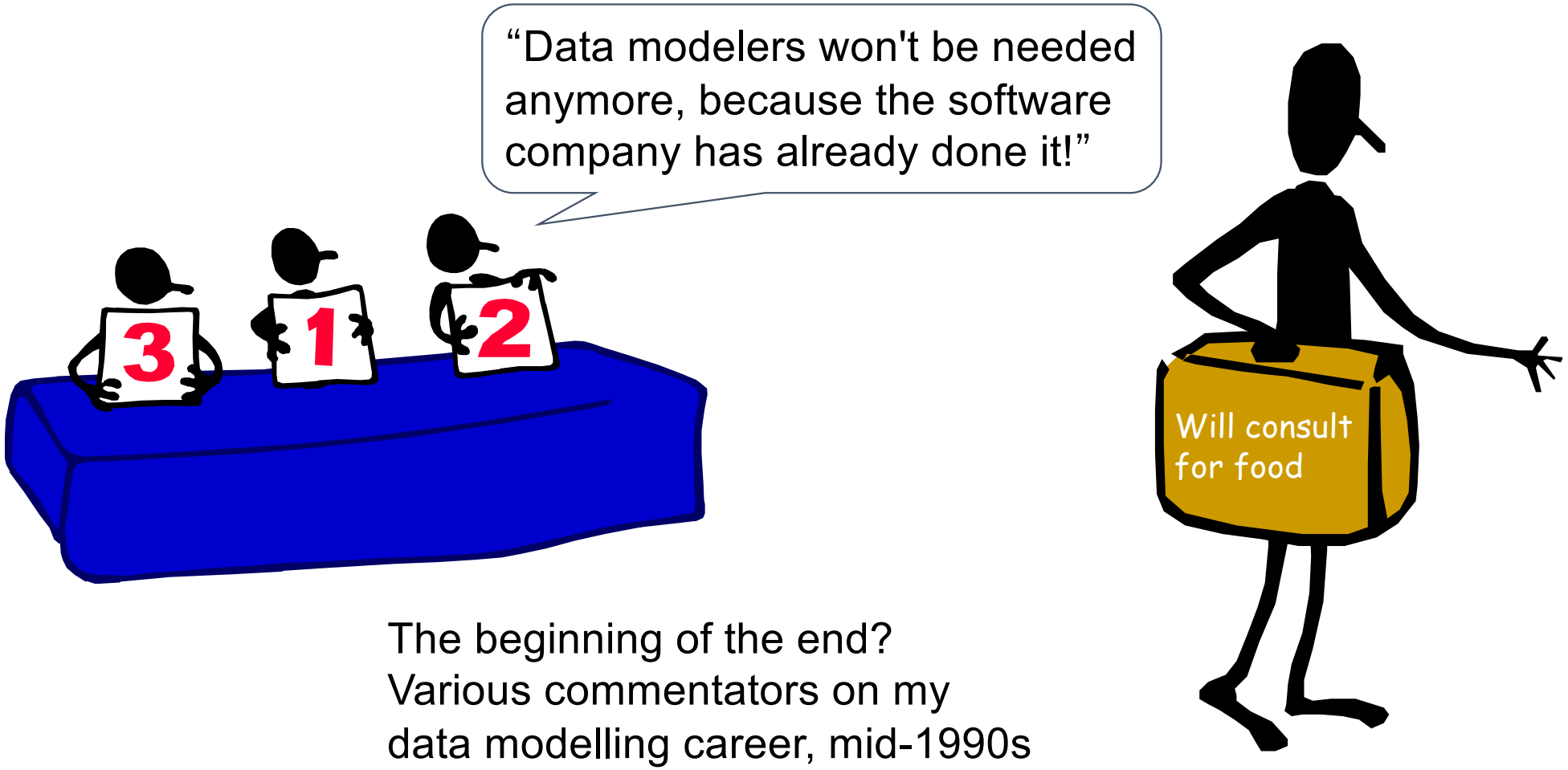
Example – simple Concept Modelling to clarify the process

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.”



Example – Data Modelling as the basis for COTS configuration



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

Redemption!

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 that stuff on our Work Order Management System, we all felt we understood our business better than we ever had

They say it's a terrible idea and a waste of time and could you please *just stay home.*

Alec...

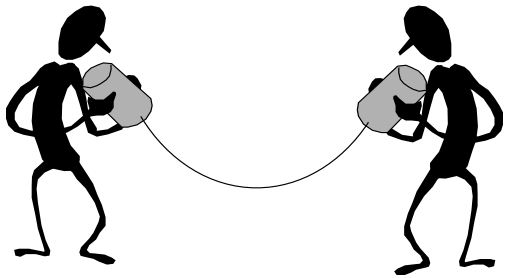
I guess. What thing in particular?

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

Uh-huh. Why do you want my help?

Great! And what do your SAP consultants say about this?

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

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!"

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.

Example: 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

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.”*

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 – Finnish architect Matti Suuronen

Initial focus – too much on "requirements"

Over 100 developers coded detailed business rules and contract terms 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

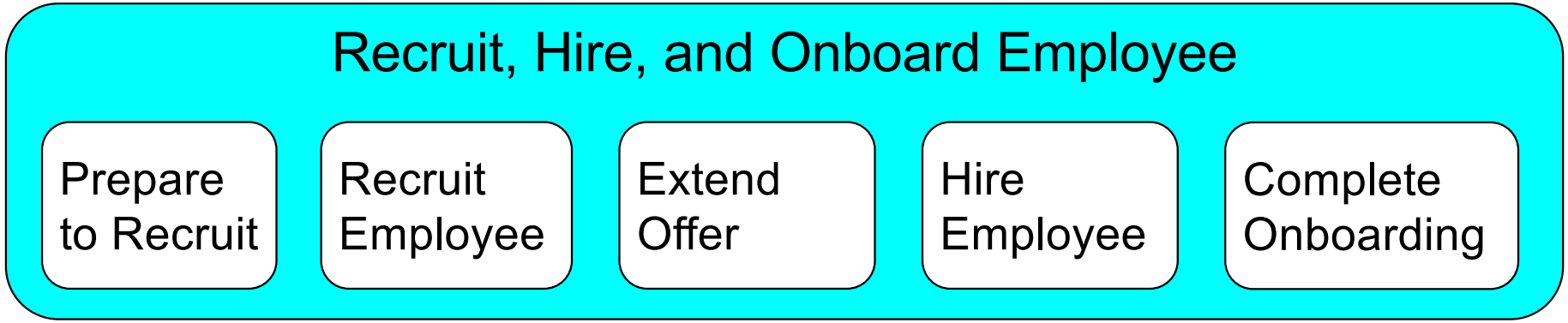
Identified, modelled, analysed, redesigned significant process – “Recruit, Hire, and Onboard Employee," the Case was "Tenure-Track Faculty”

- Developed scope model (invaluable!)
- Developed augmented scope model
- Assessed and redesigned based on “what”
- Built to-be scope model to “who – what – how” detail

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 for the full process

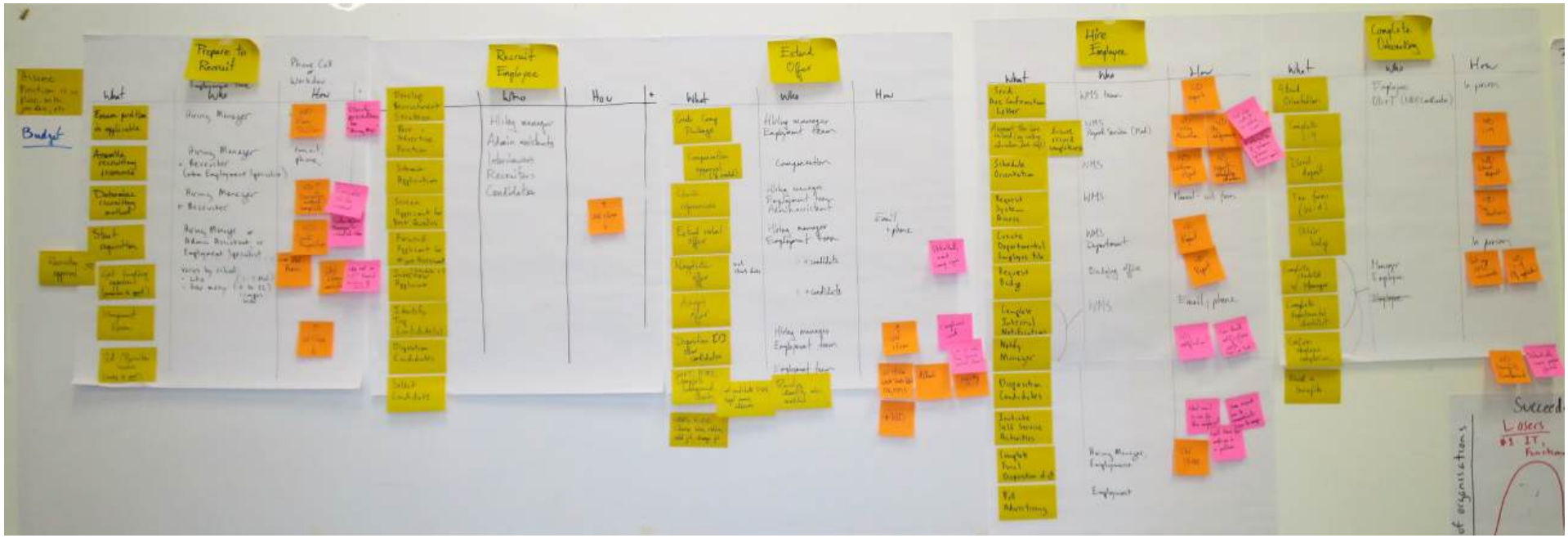
Recruit, Hire, and Onboard Employee

- *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 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 "*active verb* + noun" (They did a great job!)
- *This is critical to build support for change – it “depersonalises” in a good way!*

Then add "who and how"



Add “who” (which role) and “how” (which tool or system function) and "notes."
This format is far easier than a swimlane diagram, which would be *overwhelming!*

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!

A challenging example for your review – "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)

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.

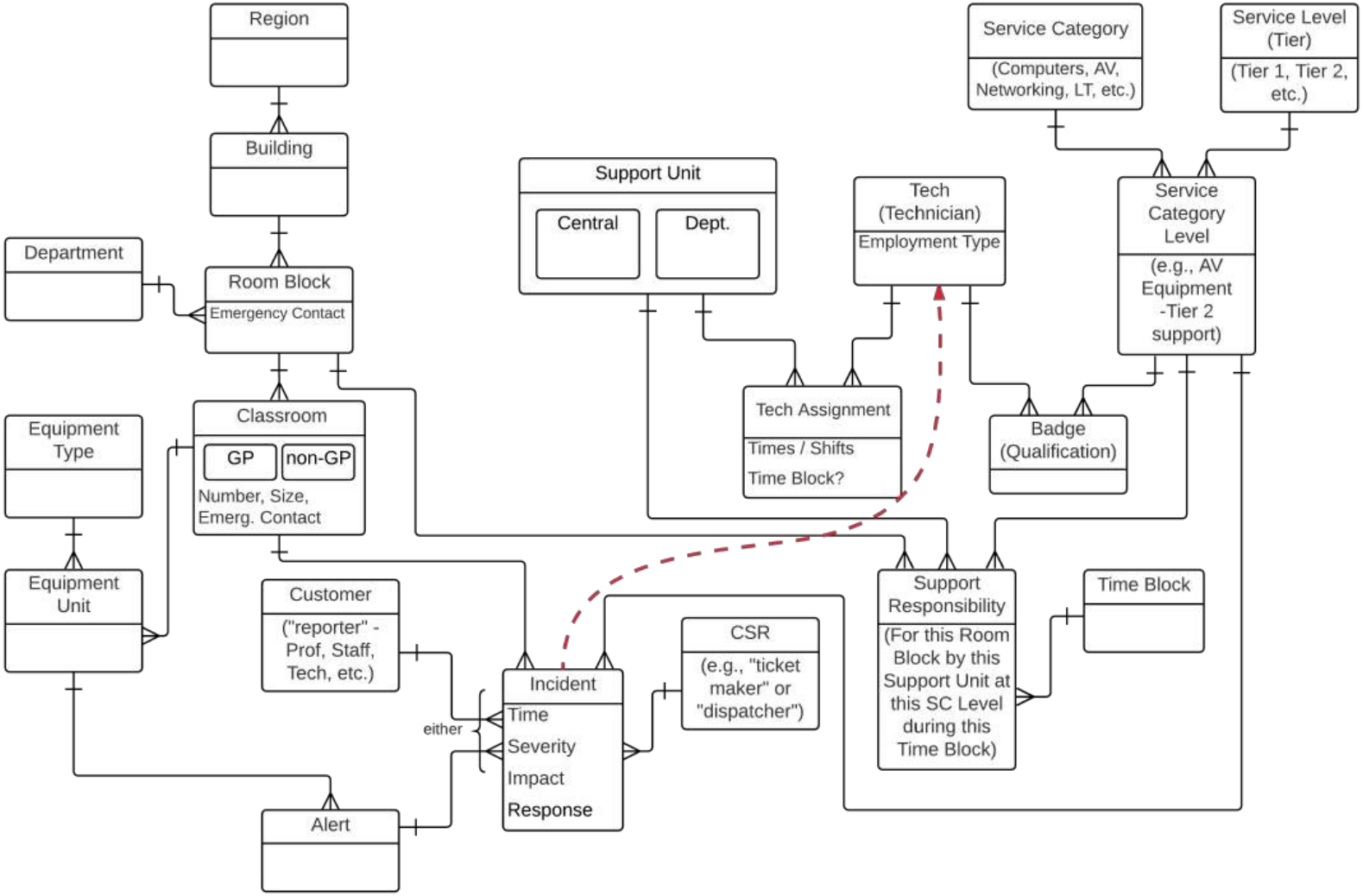
Putting all this to work…

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…

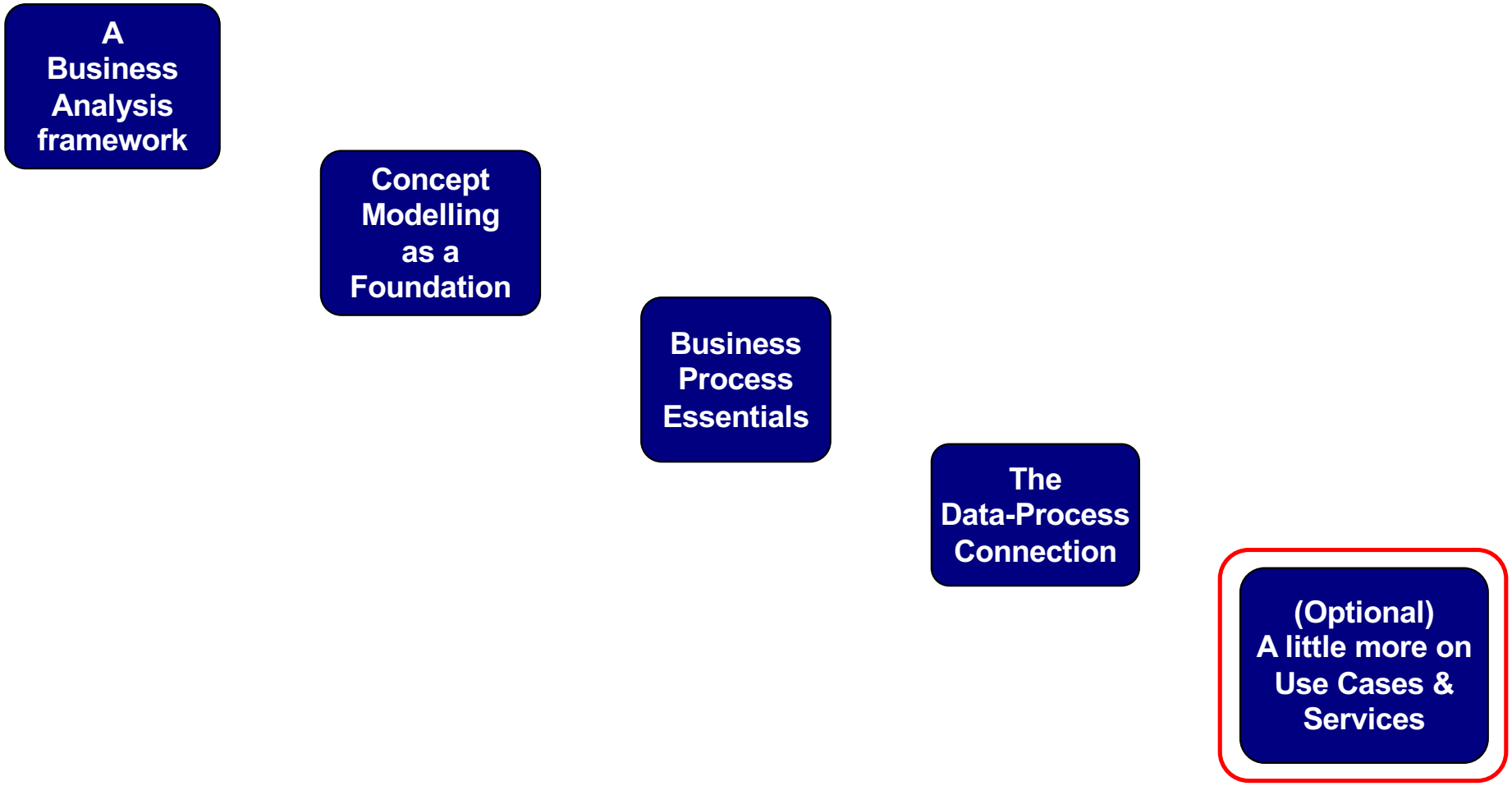
- When an Incident is raised, we know the Room Block (via Room,) the Time Block, and the Service Category Level, therefore we know the Support Responsibility, and therefore the Support Unit.
- We also know which Techs are badged for that Service Category Level, and which Techs are assigned to that Support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

Sorry about the fine print. And, no, this was not a simple job. It took some real effort to build the enabling concept model, but *we could not have done it without the assertions* – they made the needs granular!

The underlying "Conceptual Plus" Model



Overview of topics



List-based requirements – Use Cases to the rescue?

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

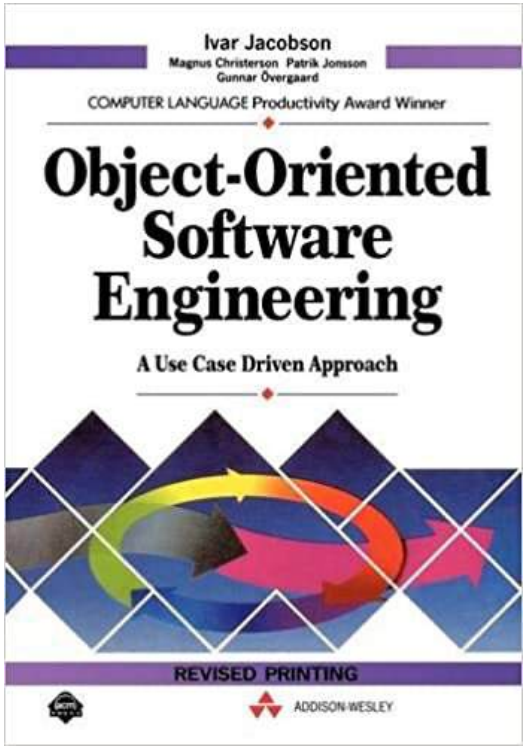
The idea – appealing in its simplicity

- Recognizable *tasks* provide *context.*

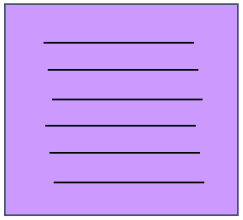
- *"Use cases are wonderful but confusing."* Alistair Cockburn
- *"A use case seems to be anything anyone wants it to be."* Charlie MacLachlan

The reality – plenty of grief and confusion

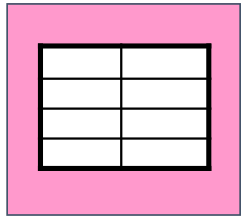
- Granularity, form, content, perspective, used for, …?
- How many use cases?
- Complete, self-contained methodology?
- Excessive complexity of some approaches



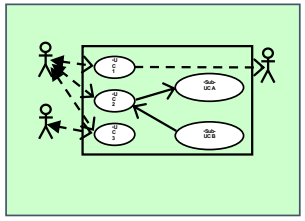
- Will the real Use Case please stand up? -



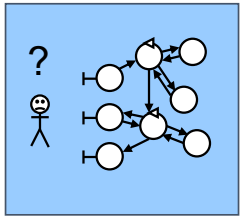
Meandering Narrative



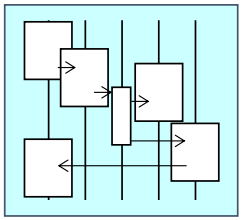
Tabular



UML Use Case



Ideal Object Model



Sequence Diagram

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

Process

Business Process

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

Application

Application requirements

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

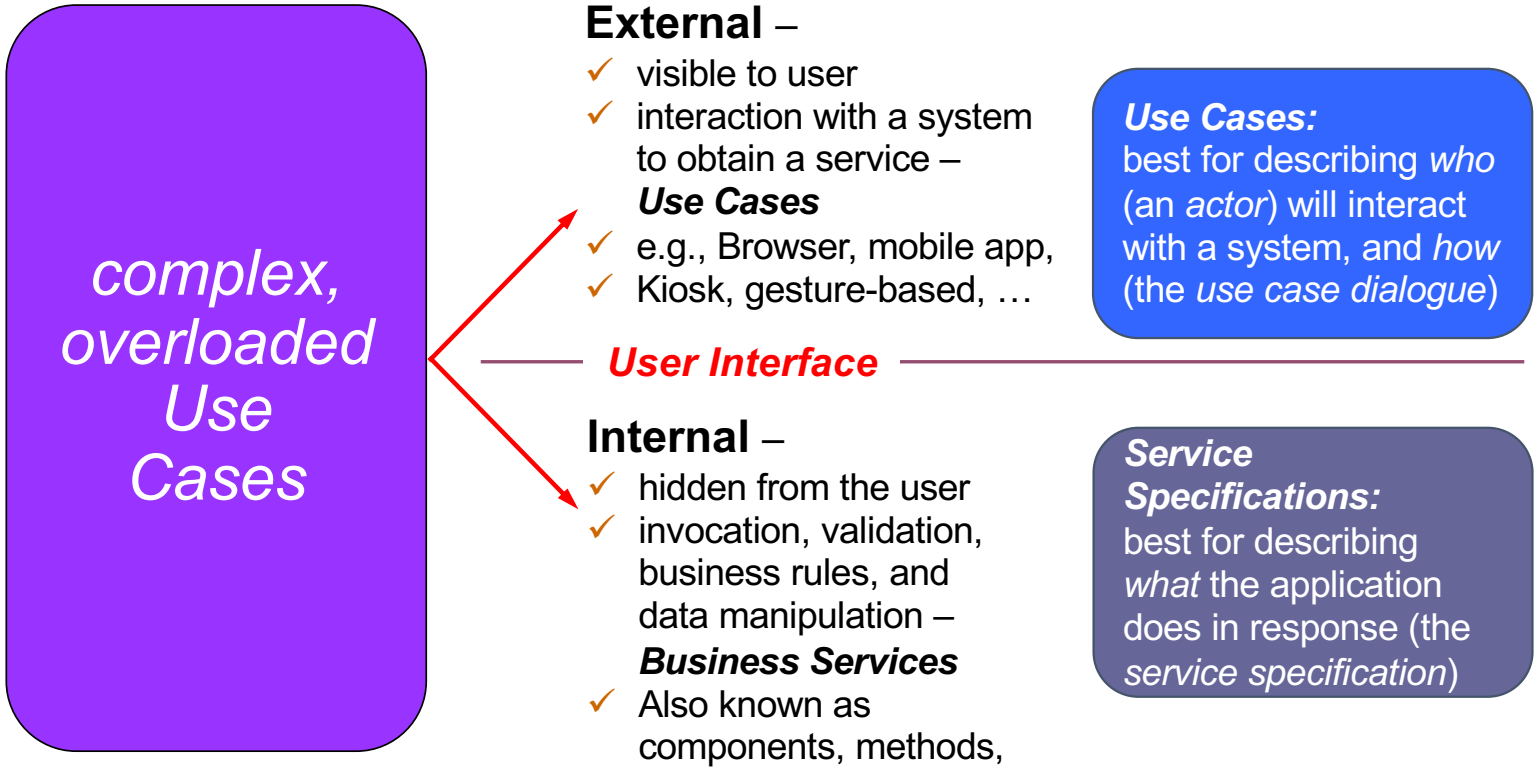
Data

Data Management

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

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

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



Services, Use Cases, Use Case Scenarios

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

Business Service:
Place Order

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

Use Case:
Customer Places Order via Web

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

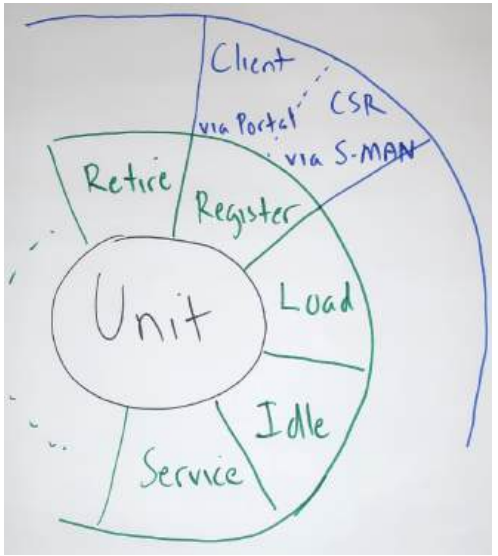
Use Case Scenario:
Joe Bloggs, a Platinum customer, places a complex order involving four ship-to addresses…

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

Discussion – one Business Service, one or more Use Cases

One Service

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



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

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

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

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

Use Cases at an ATM

We'll use the ATM as an example –
it's familiar and demonstrates key principles.

1) What actors interact with an ATM?

2) What Services would each of the actors like to access at an ATM?

Customer

-

-

-

…

Use Cases at an ATM

We'll use the ATM as an example –
it's familiar and demonstrates key principles.

1) What actors interact with an ATM?

2) What Services would each of the actors like to access at an ATM?

Customer –

1. Withdraw Funds
2. View Balance
3. Deposit Funds
4. Transfer Funds
5. Change PIN (or Call Tech Support)

Bank Employee (Teller) – Replenish Supplies

Bank Manager – Add/Drop Access Permission

Technician – Update Software

Service Provider (Brinks) – Replenish Cash

Core Banking System – Check Connection, Stop/Start ATM

Others? – YES! Manufacturer, Regulator, Other Bank, Scammer/Thief

Customer

Bank Employee (Teller)

Bank Manager

Technician

ATM Service Provider (Brinks)

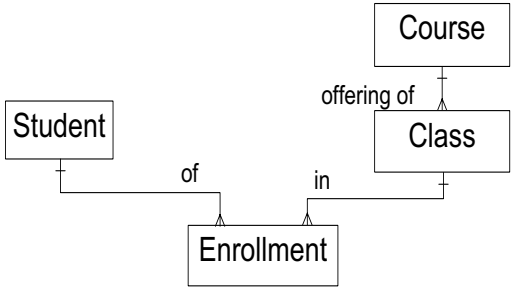
Core Banking System

…?

Initial (concept level) service specification

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

Initial (concept level) service spec example



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

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

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

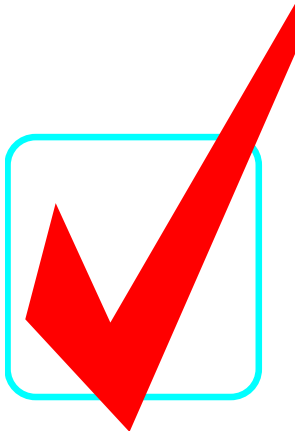
Discussion – build an Initial Service Spec

	Component	Description
1	Name	Transfer Funds (or "Complete Transfer Transaction")
2	Result	Moves a specified amount of money from one of a Customer's Accounts to another of the same Customer's Accounts, either immediately or at a future date and time.
3	Action	• Confirm Customer & Account existence and status; • Ensure Transfer Amount is within limits; • Confirm adequate balance in "From" Account; • Move funds "From" to "To"; • Generate Confirmation Number. • Update "From" and "To" Account Balances:
4	Questions & Assumptions	• Same Currency or Cross-Currency?

Intent – Determine if the Client and the Analyst share the same basic understanding.

- Anything important missed? (e.g., Calculate Fees, Eligible Accounts…)
- In Agile terms, the points are "Placeholders for a conversation."

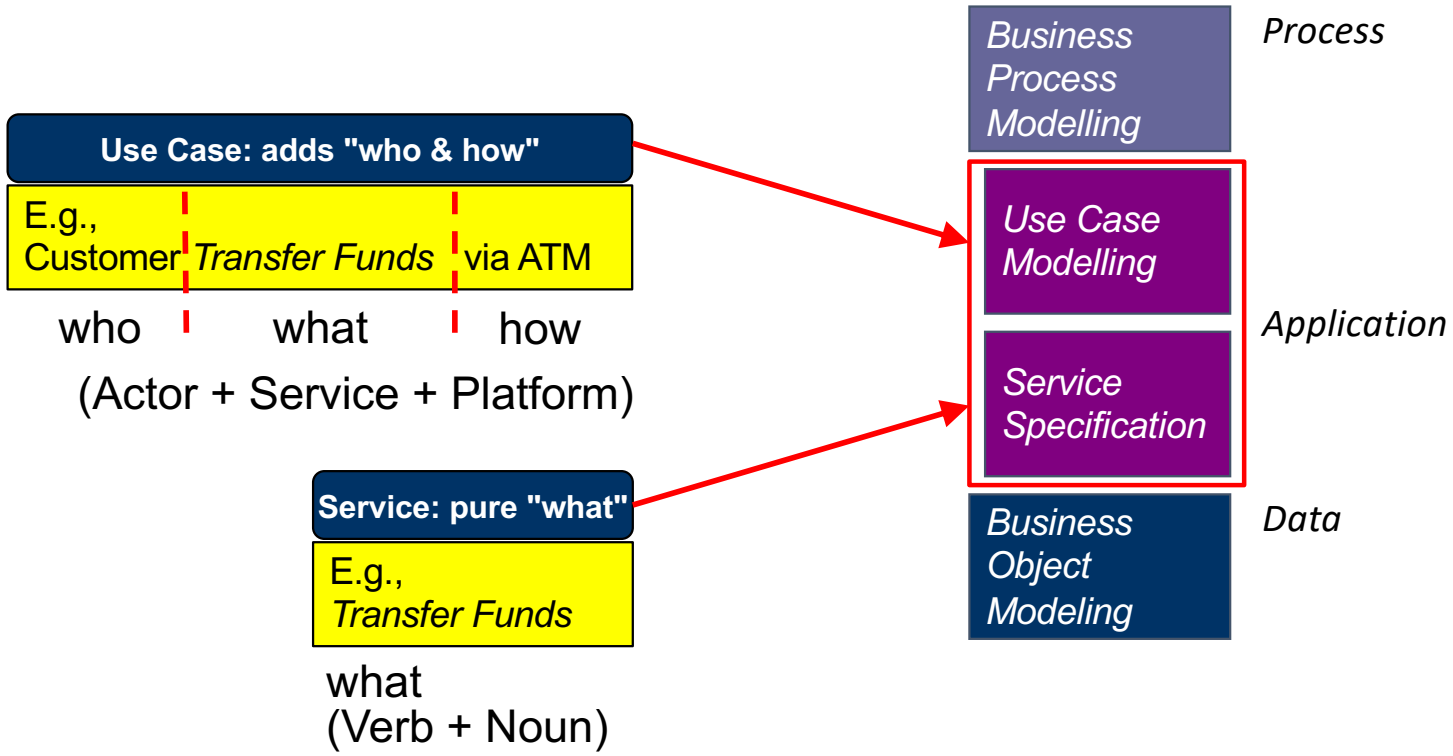
Checkpoint



Whether you did Concept or Detail level Service Specifications, you now have an excellent platform for use case development:

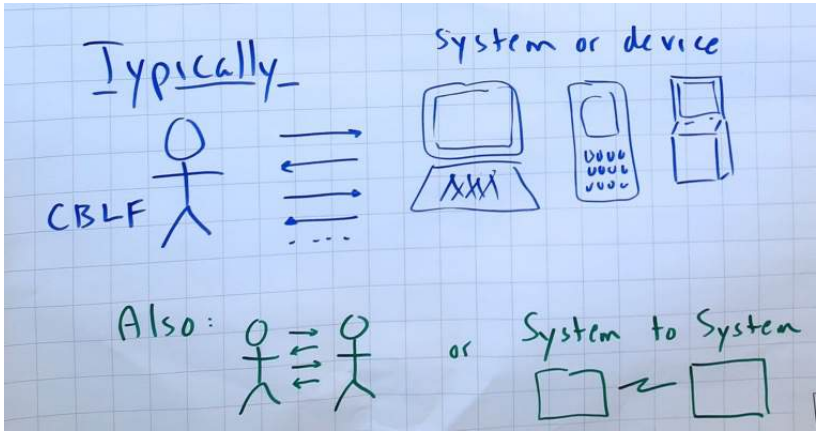
- Granularity
- Expected result
- Validation and actions

A reminder – add "who" & "how" to Services to ID Use Cases

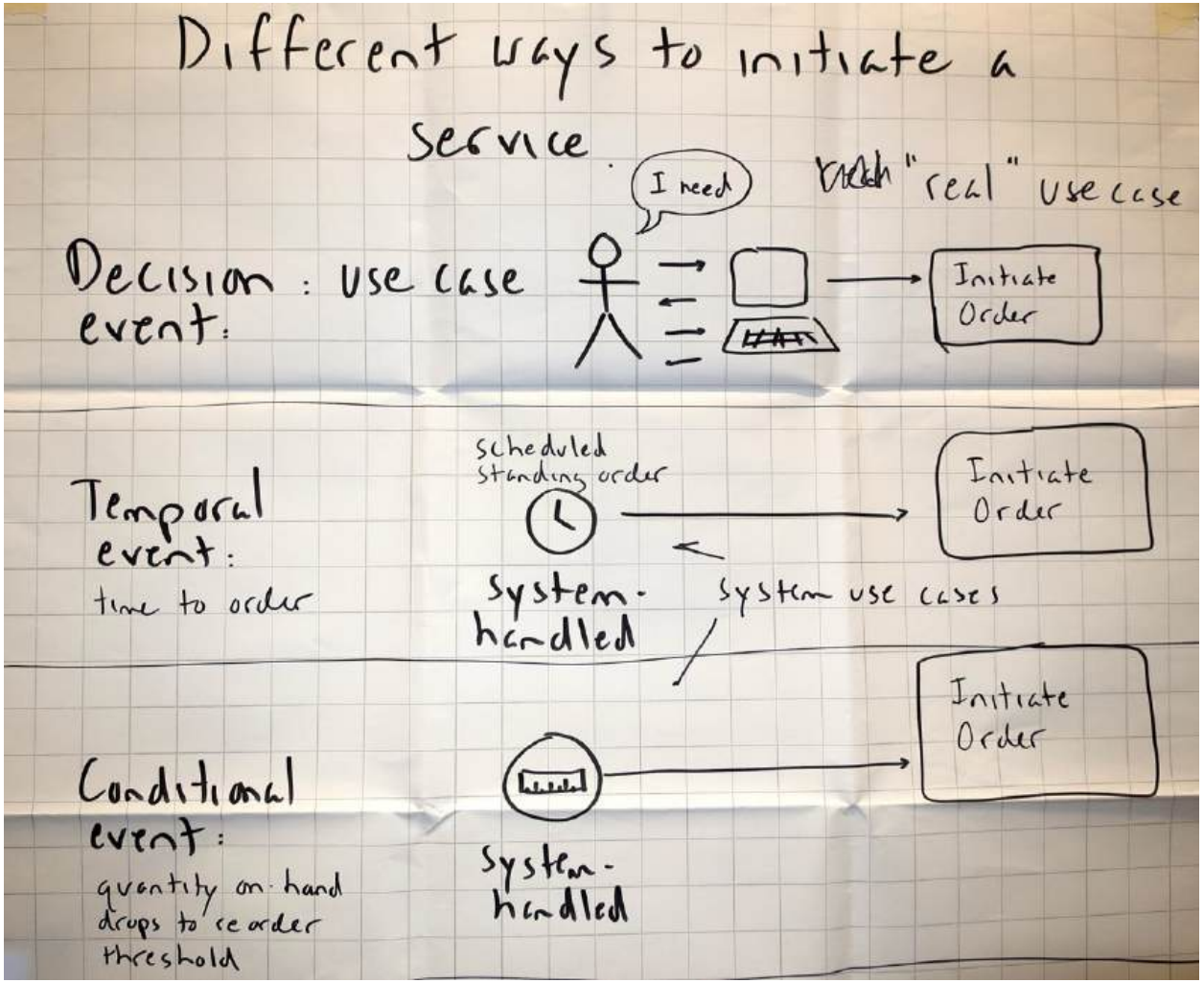


Clarifying "Use Case"

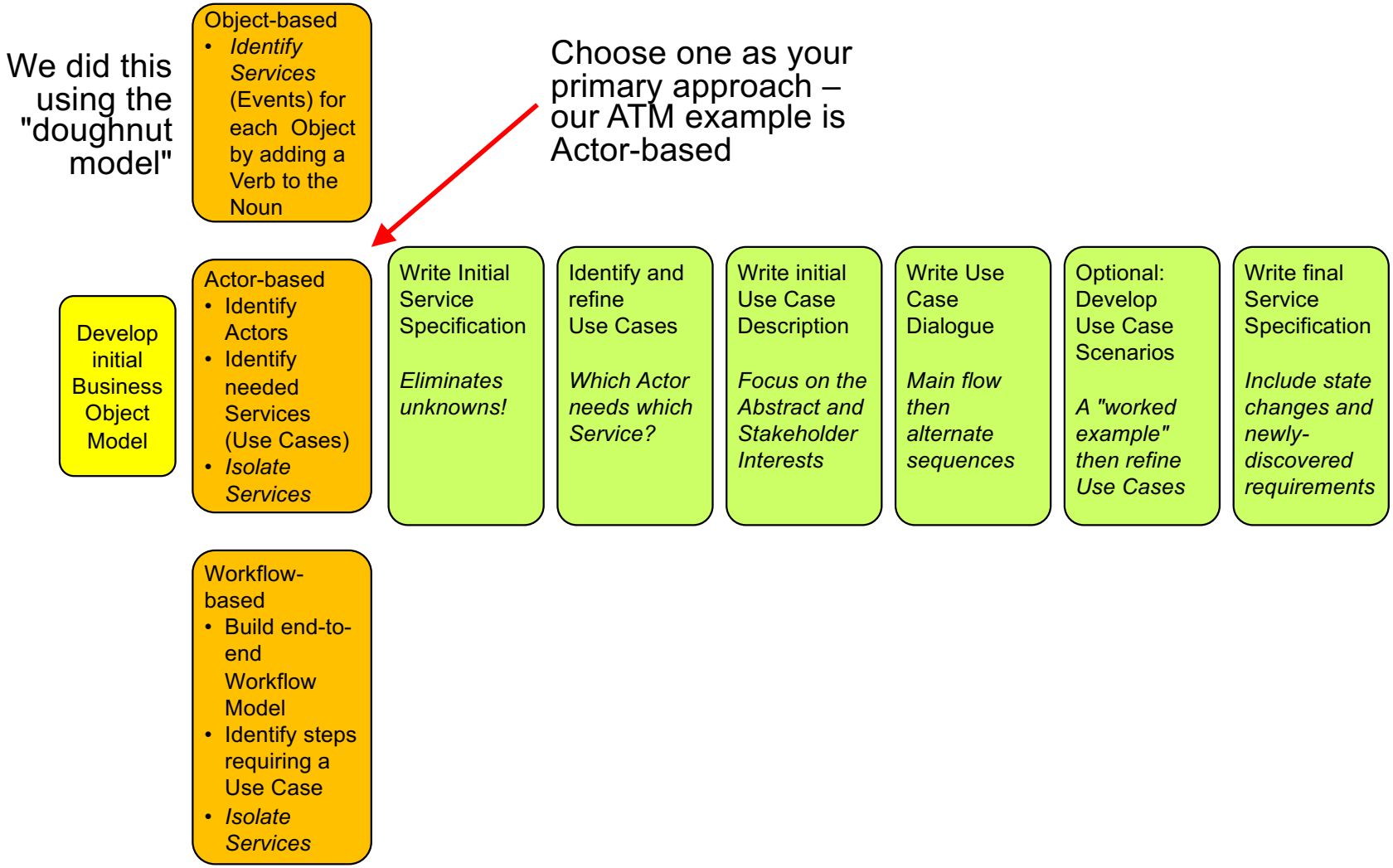
- The term “Use Case” has been overused and misused since Ivar Jacobsen first created it,
  E.g. Gartner Group –
  "Top Five Use Cases for Robotic Process Automation"
- What do *we* mean by a Use Case?
  In general, a Use Case is a description of how an Actor would like to interact with a System to complete a Task:
  - obtain a service
  - receive information
- Typically…
  - a person interacting with a system – a "Real Use Case"
  - interaction within/between systems – a "System Use Case"



"Real" Use Cases and "System" Use Cases



An approximate sequence for Services and Use Cases



Initial (concept level) use case description – 1

	Component	Description	Notes
1	**Actor(s)**	Name of primary and other actors	• Could include "supporting actors" or actors working on behalf of "ultimate primary actor" • Could be primary plus others, or "role" name
2	**Service + Platform (opt.)**	Completes the use case name	• Often specifies implementation technology or platform, but not always. • E.g. Student Complete Enrollment via Web
3	Actor goal (optional)	What the actor actually wants to achieve	• Usually is self-evident from the use case name, but may need further elaboration • Optional step
4	**Stakeholder interests**	Conditions to be met in the interest of other stakeholders not directly involved in the use case	• Ask "What would make the stakeholder unhappy?", and reverse it • Consider the customer (actor,) *other* customers, owners and managers, regulators, suppliers, … • Uncovers new rules (requirements)
5	**Abstract**	A narrative describing what happens, focusing on the main success case, in 5 +/- 2 sentences	• Provides a "mental image" or "word picture" for the entire user experience • In some cases, this is all the business analyst produces – may be done first • Also useful as a standalone product to convey the essence of the use case (e.g., in a catalog) independently of detailed documentation
6	**Notes**	Additional points or questions	• Includes questions, specific functional requirements, non-functional requirements (e.g. "dirty noisy environment") or anything noteworthy

Initial (concept level) use case description – 2

Component	Description	Notes
7 Trigger (optional)	• The action that starts the use case	• E.g., "Customer phones Sales Rep with complaint • May be the first step in the use case • Not "essential" – includes "who and how" • Optional step
8 Preconditions (optional)	• What the system can ensure is true at beginning	• E.g., "Service Rep has transaction authority" • Checked once, at the start • Not the same as all the Business Object state, data value, etc. checking that occurs later • Optional step – really only useful for out-of-context use cases
9 Milestones or steps (optional)	• Main phases or steps in ***main success case***	• Note if sequence is mandatory or optional • Alternate and failure conditions and handling will be documented later • Optional step – service specification covers it
10 Postconditions (optional)	• Important states or data values on completion	• E.g., "Complaint is recorded, and escalated if necessary" • Optional step – service specification covers it

Use Case Abstract sample – Customer Transfer Funds via ATM

May, a busy young mother, needs to quickly transfer SGD 200 from her savings account to her chequing account in order to cover a childcare payment.

She is dismayed there are several people in front of her, but the ATM deals with them efficiently and the line soon clears. Reaching the front of the line (and anxious because there is now a line behind her) she inserts her bank card to begin authorisation, then quickly specifies the "from" and "to" accounts, the amount, and confirms it is an "immediate" transfer.

Declining a receipt, she is done within 30 seconds, and carries on with her busy day, undertaking construction projects at home.

Develop initial use case dialogues

1 Develop or propose dialog, always in "when – then" format

Recap the Use Case Abstract for participants, then create dialog:

Trigger – Customer decides to contact Service Centre:

1. When Customer places service call
   Then Service Rep accepts call, greets Customer, and requests Customer Name or ID
2. When Customer provides Name
   Then Service Rep uses Customer Search, System displays search results list, and Service Rep requests confirmation
3. When Customer provides confirmation
   Then Service Rep requests details of the issue or problem…

- ✓ Each clause (step) goes from the "driving" actor having control to that actor being given control again
- ✓ Ultimately, the dialogue must:
  - be satisfying for the actor
  - protect stakeholder interests
  - be valid against object model dependency and rules
  - enforce service rules and provide the data it needs
- ✓ KEY POINT – it's just fine to give a brief indication of "behind the scenes" system activity – it provides context!

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. *Our most popular workshop!*

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.

Advanced Business Process Techniques – *Aligning Process Work with Strategic, Organisational, and Cultural Factors* 2 days

We regularly hear about the importance of "alignment" in achieving success when working with business processes, but alignment with what? This workshop provides specific, repeatable techniques to help your business process initiatives align with human factors, organisational culture, and enterprise strategy and goals. Rather than save these concerns for a single "think about people, culture, and strategy" phase, it shows how to incorporate them at every stage, from process identification, scoping, and initial assessment through to modelling, analysis, and resign. Regularly receives rave reviews.

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

This highly interactive workshop combines the core content from two popular data modelling offerings by Alec Sharp – "Business Oriented Data Modelling" and "Advanced Data Modelling." The first day of the workshop gets both new and experienced modellers to the same baseline on terminology, conventions, and unique, business-engaging approaches. The next two days 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. In all cases, the underlying philosophy is that a data model is a description of a business, not of a database.

Three main themes are explored in a very practical way:

1. The foundations of data modelling – what a data model *really* is, and maximising its relevance
2. The human side of data modelling - improving communication skills and engaging the business
3. The complex side of data modelling - getting better at modelling difficult situations

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!

CLARITEQ

Alec Sharp, West Vancouver, BC, Canada

If you have questions or comments…
don't be shy, get in touch!

- e: asharp@clariteq.com
- t: @alecsharp
- ig: @alecsharp01
- m: +1 604 418-3352