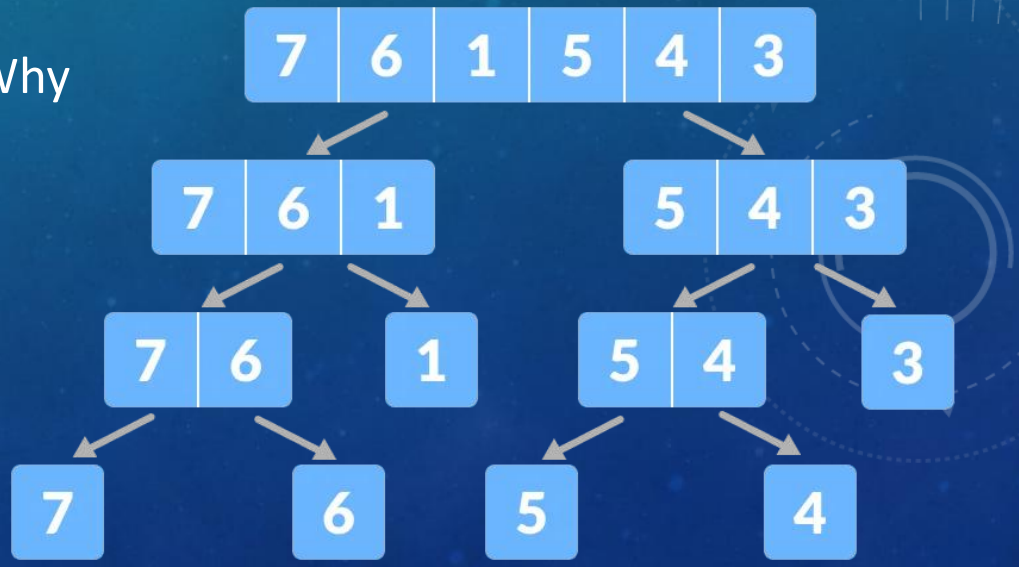
REQUIREMENTS ANALYSIS

PART ONE: MODELLING (CH8)

USER STORIES

User stories

- A user story is a “short, simple description of **a** feature told from the perspective of the person who desires the new capability, usually a user or customer of the system”
- ”Standard” format of user story is: *As a <type of user>, I want <a goal> so that <some reasons>.*
- This contains the following key information:
  - Role of the user playing in a system <type of user> -- who
  - Goals to achieve <a goal> -- what
  - The reasons for achieving the goal <some reasons> -- Why
- Process of establishing user stories (divide and conquer)
  - Start from a big goal
  - Split the goal into smaller



USER STORIES

- Agile software development uses user stories to represent user requirements.
- A user story will be broken down (refined) into detailed and smaller ones. Normally a user story should be small enough to fit in one iteration in Agile , so an acceptance test is carried out at the end of the iteration.
- Some examples of user stories:

Application	Sample use case	Corresponding user story
Chemical tracking system	Request a Chemical	As a chemist, I want to request a chemical so that I can perform experiments.
Airport check-in kiosk	Check in for a Flight	As a traveler, I want to check in for a flight so that I can fly to my destination.
Accounting system	Create an Invoice	As a small business owner, I want to create an invoice so that I can bill a customer.
Online bookstore	Update Customer Profile	As a customer, I want to update my customer profile so that future purchases are billed to a new credit card number.

USE CASES

- Use case, represented either with a table or use case diagrams and activity diagrams, provides a high-level visual representation of users' operations from which the user requirements can be drawn.
- Elements:
  - An actor is a person (or sometimes another software system or a hardware device) that interacts with the system to perform a use case
  - A use case might encompass a number of related activities leading to a goal. A scenario is a description of a single instance of the usage of the system. A use case is therefore a collection of related usage scenarios, and one scenario is a specific instance of a use case.
- identify actors:
  - Who (or what) is notified when something occurs within the system?
  - Who (or what) provides information or services to the system?
  - Who (or what) triggers the system to start a task?

USE CASES

- Identifying use cases in several ways:
  - Identify the actors first, then lay out the business processes being supported by the system secondly, and then define the use cases for activities where actors and systems interact.
  - Create specific scenarios to illustrate a business process, then generalise the scenarios into use cases and identify the actors involved in each one.
  - Using a business process description, ask, “What tasks must the system perform to complete this process or convert the inputs into outputs?” Those tasks might be the use cases.
  - Identify the external events to which the system must respond, then relate these events to participating actors and specific use cases.
  - Use CRUD analysis to identify data entities that require use cases to create, read, update, delete, or otherwise manipulate them.
  - Examine the context diagram and ask, “What objectives do each of these external entities want to achieve with the help of the system?”

USE CASES

- Use cases can be represented in either table or activity diagram, and they must have the following elements:
  - A unique identifier and a succinct name that states the user goal
  - A brief textual description that describes the purpose of the use case
  - A trigger condition that initiates the execution of the use case
  - Zero or more preconditions that must be satisfied before the use case can begin
  - One or more postconditions that describe the state of the system after the use case is successfully completed
    - Something observable to the user (the system displayed an account balance).
    - Physical outcomes (the ATM has dispensed cash and printed a receipt).
    - Internal system state changes (the account has been debited by the amount of a cash withdrawal, plus any transaction fees).
  - A numbered list of steps that shows the sequence of interactions between the actor and the system—a dialog—that leads from the preconditions to the postconditions

USE CASES

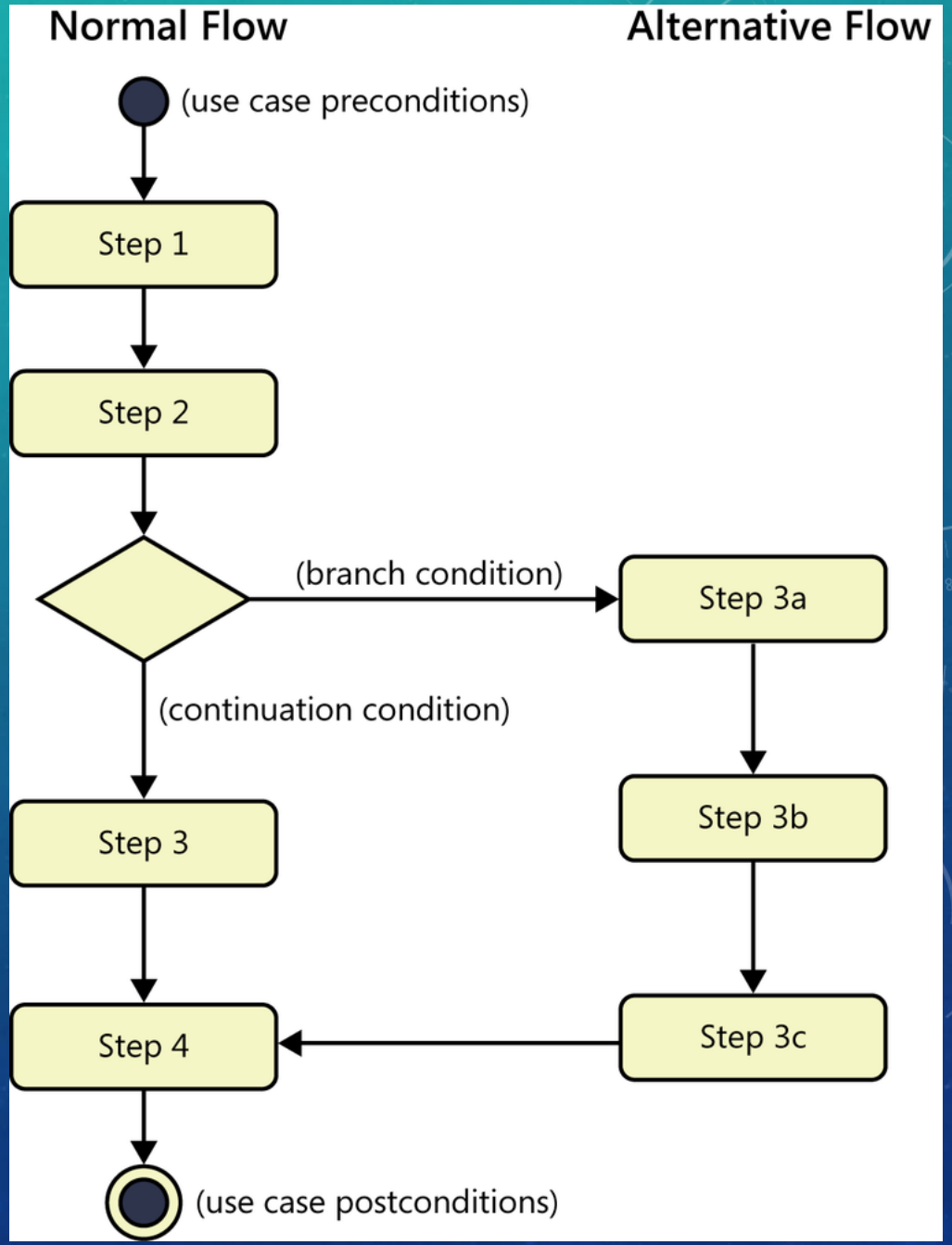
- Table example
- (Exercise: convert the table to use case diagram and activity diagram in UML?)

Action schema:
Precondition+action+postcondition.
Precondition is sufficient for action
Postcondition is the consequence

ID and Name:	UC-4 Request a Chemical		
Created By:	Lori	Date Created:	8/22/13
Primary Actor:	Requester	Secondary Actors:	Buyer, Chemical Stockroom, Training Database
Description:	The Requester specifies the desired chemical to request by entering its name or chemical ID number or by importing its structure from a chemical drawing tool. The system either offers the Requester a container of the chemical from the chemical stockroom or lets the Requester order one from a vendor.		
Trigger:	Requester indicates that he wants to request a chemical.		
Preconditions:	PRE-1. User's identity has been authenticated. PRE-2. User is authorized to request chemicals. PRE-3. Chemical inventory database is online.		
Postconditions:	POST-1. Request is stored in the CTS. POST-2. Request was sent to the Chemical Stockroom or to a Buyer.		
Normal Flow:	**4.0 Request a Chemical from the Chemical Stockroom** 1. Requester specifies the desired chemical. 2. System lists containers of the desired chemical that are in the chemical stockroom, if any. 3. System gives Requester the option to View Container History for any container. 4. Requester selects a specific container or asks to place a vendor order (see 4.1). 5. Requester enters other information to complete the request. 6. System stores the request and notifies the Chemical Stockroom.		
Alternative Flows:	**4.1 Request a Chemical from a Vendor** 1. Requester searches vendor catalogs for the chemical (see 4.1.E1). 2. System displays a list of vendors for the chemical with available container sizes, grades, and prices. 3. Requester selects a vendor, container size, grade, and number of containers. 4. Requester enters other information to complete the request. 5. System stores the request and notifies the Buyer.		
Exceptions:	**4.1.E1 Chemical Is Not Commercially Available** 1. System displays message: No vendors for that chemical. 2. System asks Requester if he wants to request another chemical (3a) or to exit (4a). 3a. Requester asks to request another chemical. 3b. System starts normal flow over. 4a. Requester asks to exit. 4b. System terminates use case.		
Priority:	High		
Frequency of Use:	Approximately 5 times per week by each chemist, 200 times per week by chemical stockroom staff		
Business Rules:	BR-28, BR-31		
Other Information:	The system must be able to import a chemical structure in the standard encoded form from any of the supported chemical drawing packages.		
Assumptions:	Imported chemical structures are assumed to be valid.		

USE CASES

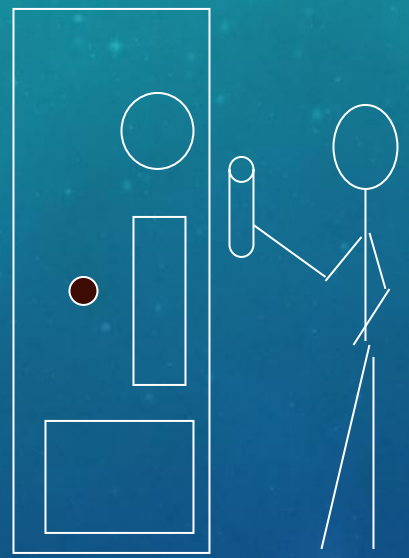
- Activity diagram
  - One scenario is identified as the *normal flow* of events for the use case. It's also called the main flow, basic flow, normal course, primary scenario, main success scenario, sunny-day scenario, and happy path.
  - Other success scenarios within the use case are called *alternative flows* or *secondary scenarios*. Alternative flows deliver the same business outcome (sometimes with variations) as the normal flow but represent less common or lower-priority variations in the specifics of the task or how it is accomplished.



USE CASES

- Example and exercise:

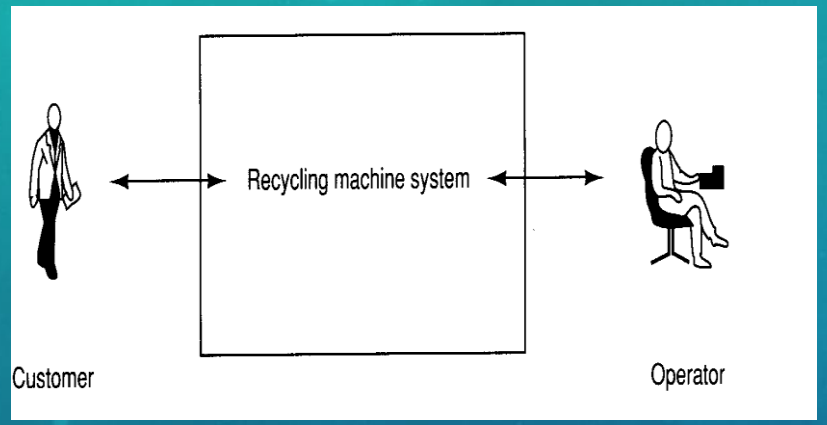
  System description: The system controls a recycling machine for returnable bottles, cans and crates. The machine can be used by several customers at the same time and each customer can return all three types of item on the same occasion.



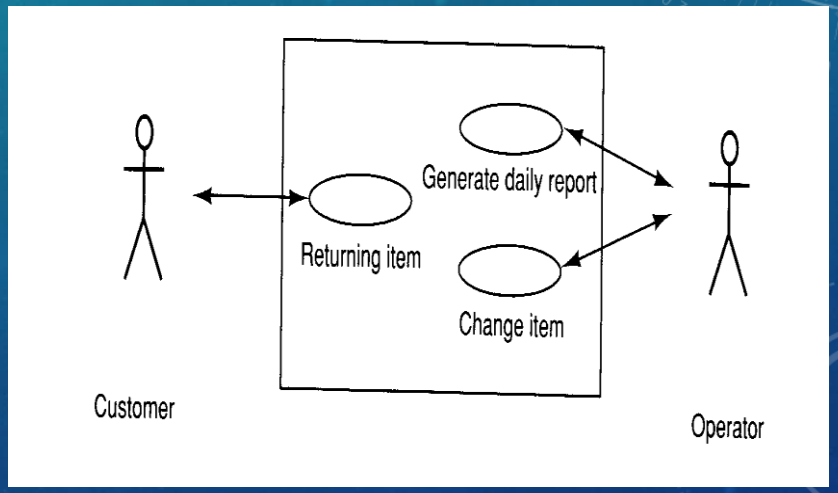
The current business process:
The **counter** has to check, for each item, what type has been returned. He will register how many items each **customer** returns and when the customer asks for a receipt, the counter will print out what was deposited , the value of the returned items and the total return sum that will be paid to the customer. A **manager** monitors the counter. He asks for a printout of the total number of items that have been deposited at the end of each day. He has right to **change** the deposit values of the items through a console. When anything wrong with the process, the manager will be called by the counter by sending a special alarm signal.

USE CASES

- Identify actors who are outsiders of the recycling process but interact to it



- Identify use cases that representing sequences of activities



Returning items

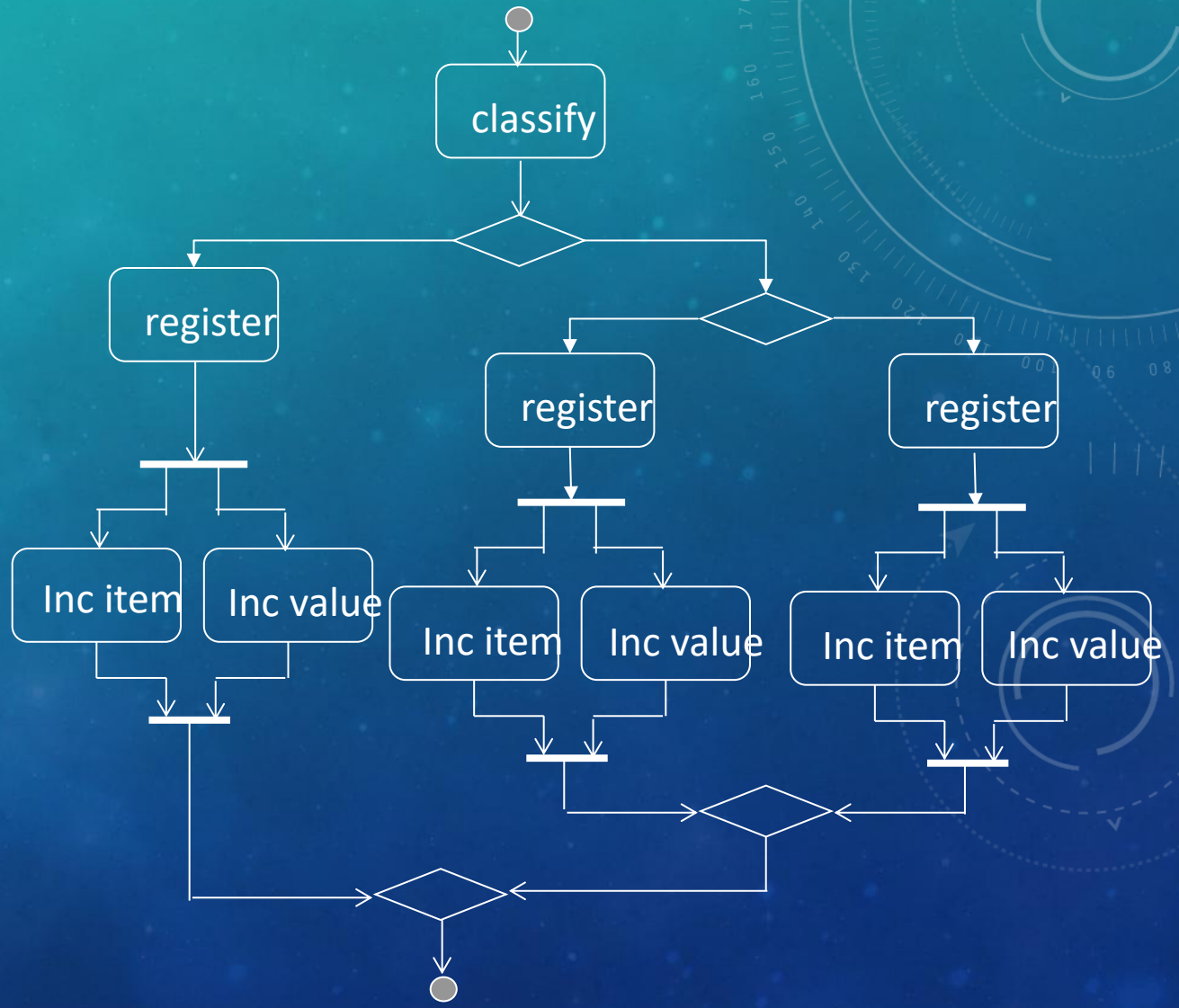
Generating daily report

Changing item values

USE CASES

- Activity diagram provides detailed activities within use cases
- focusing on return item use case

  When a customer returns an item to the machine, the following activities should happen in the system:
  - Register the item
  - Classify (crate | can | bottle)
  - Increment item quantity
  - Increment item value.



PART TWO: BUSINESS RULES AS CONSTRAINS (CH9)

BUSINESS RULES

Definition

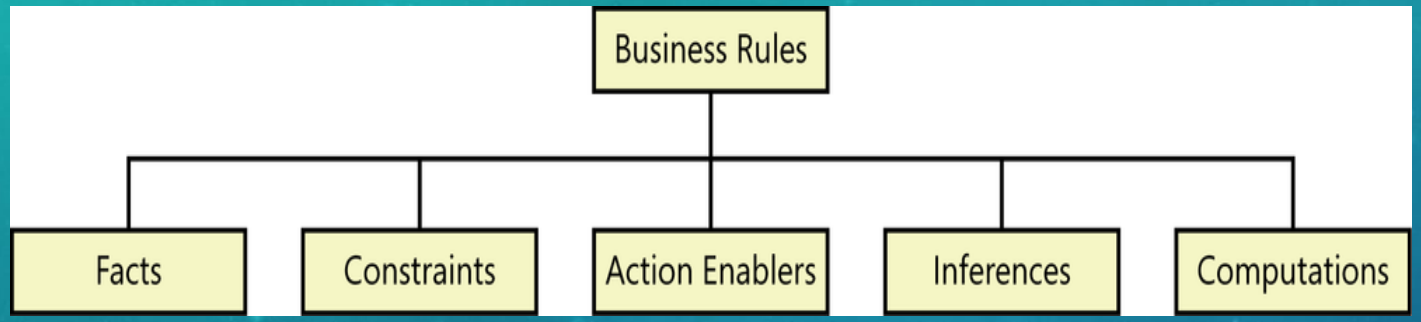
- Software must follow policies, laws, and industry standard and government regulations. All of these are called business rules
- Definitions from the perspectives of both the business and its information systems:
  - From the business perspective: “A business rule is the guidance indicating an obligation concerning conduct, action, practice, or procedure within a particular activity or sphere.” (There ought to be an explicit motivation for the rule, as well as enforcement methods and an understanding of what the consequences would be if the rule were broken.)
  - From the information system perspective: “A business rule is a statement that defines or constrains some aspect of the business. It is intended to assert business structure or to control or influence the behaviour of the business.”
- Examples? (Agresso – expenditure refund procedure requires authorisation from budget holder, line manager and finance, and double check by salaries.)

BUSINESS RULES

Inference in predicate logic:
Premise 1: A
Premise 2: If A then B

Conclusion: B

A business rules taxonomy (How business rules are classified/organised)



- Facts are simply statements that are true about the business at a specified point in time. A fact describes associations or relationships between important business terms. Example:
  - Every order has a shipping charge. (predicate)
- Action enabler that triggers some activities if specific conditions are true. Example:
  - If the customer ordered a product of certain, then offer the customer the related product before completing the order (we all experienced this from Amazon/Jingdong/……) (?)
- An inference creates a new fact from other facts, often in the form of "if/then". (Difference to action-enabling business rules, in the way that the "then" clause of an inference simply provides a piece of knowledge, not an action to be taken. Example:
  - If a payment is not received within 30 calendar days after it is due, then the account is delinquent. (rule)

BUSINESS RULES

- Computations transform existing data into new data by using specific mathematical formulas or algorithms. (actually, computation is a combination of a set of fact and a set of inferences)
- Examples
  - The total price for an order is the sum of the price of the items ordered, less any volume discounts, plus state and county sales taxes for the location to which the order is being shipped, plus the shipping charge, plus an optional insurance charge.
  - (Can be in other format rather than text -- The unit price is reduced by 10 percent for orders of 6 to 10 units, by 20 percent for orders of 11 to 20 units, and by 30 percent for orders of more than 20 units)

ID	Number of units purchased	Percent discount
DISC-1	1 through 5	0
DISC-2	6 through 10	10
DISC-3	11 through 20	20
DISC-4	More than 20	30

BUSINESS RULES

- A constraint is a statement that restricts the actions that the system or its users are allowed to perform. (fact?)
  - Organizational policies
    - A library patron may have a maximum of 10 items on hold at any time.
    - Any credit card transaction does not display more than four digits of the card number.
  - Government regulations
    - All software applications must comply with government regulations for usage by visually impaired persons. (You see the wheelchair symbol on websites.)
    - Airline pilots must receive at least 8 continuous hours of rest in every 24-hour period. (Flight stops in HK when flying from London to Australia.)
  - Industry standards
    - Mortgage loan applicants must satisfy the Federal Housing Authority qualification standards. (e,g. age check.)
    - Web applications may not contain any HTML tags or attributes that are deprecated according to the HTML 5 standard.

BUSINESS RULES

Document business rules

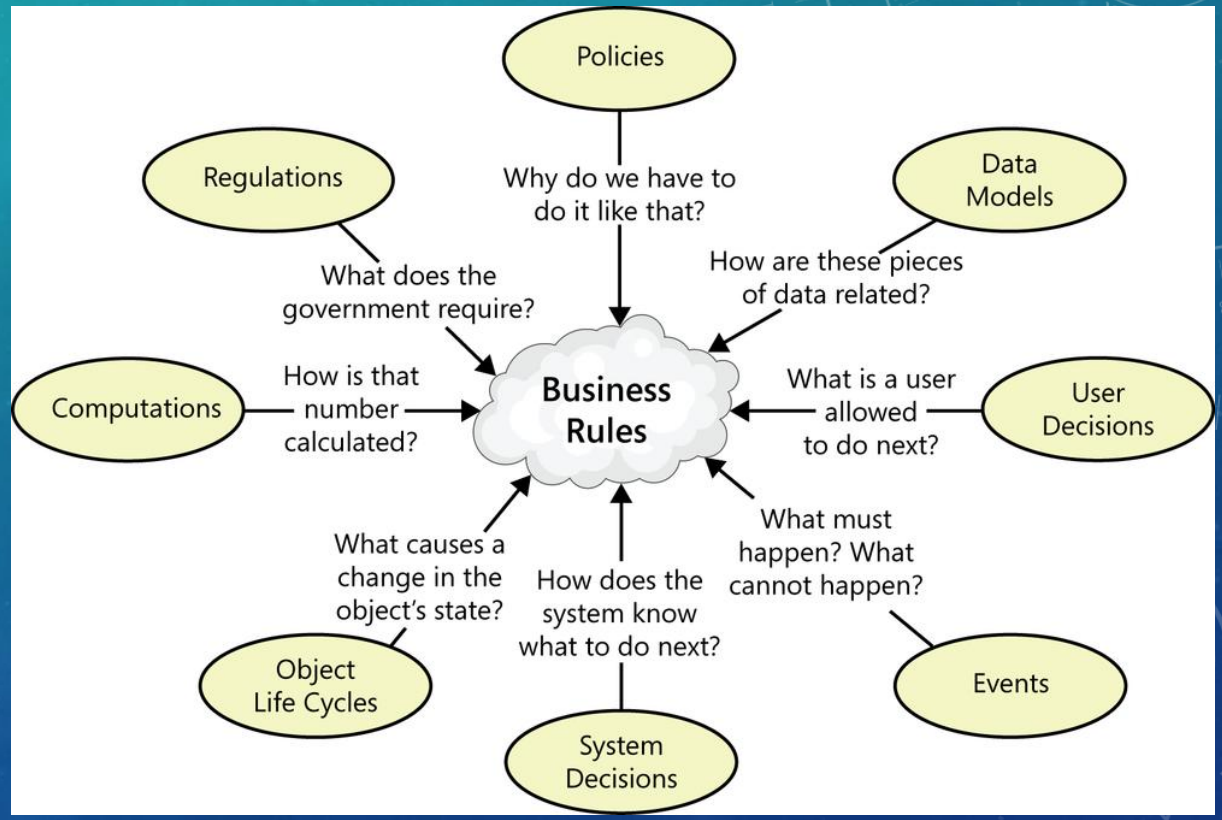
- Try the simple format

ID	Rule definition	Type of rule	Static or dynamic	Source
ORDER-5	If the customer ordered a book by an author who has written multiple books, then offer the author's other books to the customer before completing the order.	Action enabler	Static	Marketing policy XX
ACCESS-8	All website images must include alternative text to be used by electronic reading devices to meet accessibility requirements for visually impaired users.	Constraint	Static	ADA Standards for Accessible Design
DISCOUNT-13	A discount is calculated based on the size of the current order, as defined in Table BR-060.	Computation	Dynamic	Corporate pricing policy XX

BUSINESS RULES

Discover business rules

- Asking questions from different perspectives (see the diagram)
- “Common knowledge” from individuals who have worked with the business for a long time
- Legacy systems
- Business process modelling
- Existing documentation
- Compliance departments in companies as they ensure that a business adheres to external rules and internal controls
- Observation and reasoning



BUSINESS RULES

Business rules into requirements

- Consider a retail store's policy that only supervisors and managers are allowed to issue cash refunds larger than £50. If you're developing a point-of-sale application for use by store employees, this rule implies that each user must have a privilege level. The software must check to see if the current user is of sufficiently high privilege level to perform certain actions, such as opening the cash register drawer so a cashier can issue a refund to a customer.

Exercise: I went to Coop supermarket yesterday afternoon. I picked up a French bread and a bottle of milk and tried to pay with a self-service pay station. I scanned the milk but could not scan the bread. The shop assistant told me that I had to place the milk to the packaging area and then I could scan the next item.
What business rule is here?

PART THREE: REQUIREMENTS PRIORITISATION (CH16)

REASONS FOR PRIORITISATION

- Few software projects deliver all the capabilities on time and in budget. Every project with resource limitations needs to define the relative priorities of the requested product capabilities.
- Delivers the most critical or valuable functionality as early as possible (to ensure customers that their investment is in safe hands.)
- On every project, a project manager must balance the desired project scope against the constraints of schedule, budget, staff, and quality goals.
- One way to accomplish this is to drop—or to defer to a later release—low-priority requirements when new, more essential requirements are accepted or when other project conditions change.
- Prioritisation also helps reveal competing goals, resolve conflicts, plan for staged or incremental deliveries, control scope creep, in addition to make the necessary trade-off decisions.

HOW TO PRIORITISE

In or out model

- The simplest of all prioritization methods is to have a group of stakeholders work down a list of requirements and make a binary decision: is it-in, or is it-out, based on the project's business objectives. Then, reduce the list down to the bare minimum needed for the first release. Then, when implementation of that release is under way, come back to the previously "out" requirements and go through the process again for the next release.
- Pairwise comparison and rank ordering – pairwise compare requirements using a spreadsheet.

Example: Ranking requirements A, B, C, D, E

From the first row: C is more important than A

Also from the first row: A is more important than B, D and E

From the second row: B is more important than D and E

From the fourth row: D is more important than E

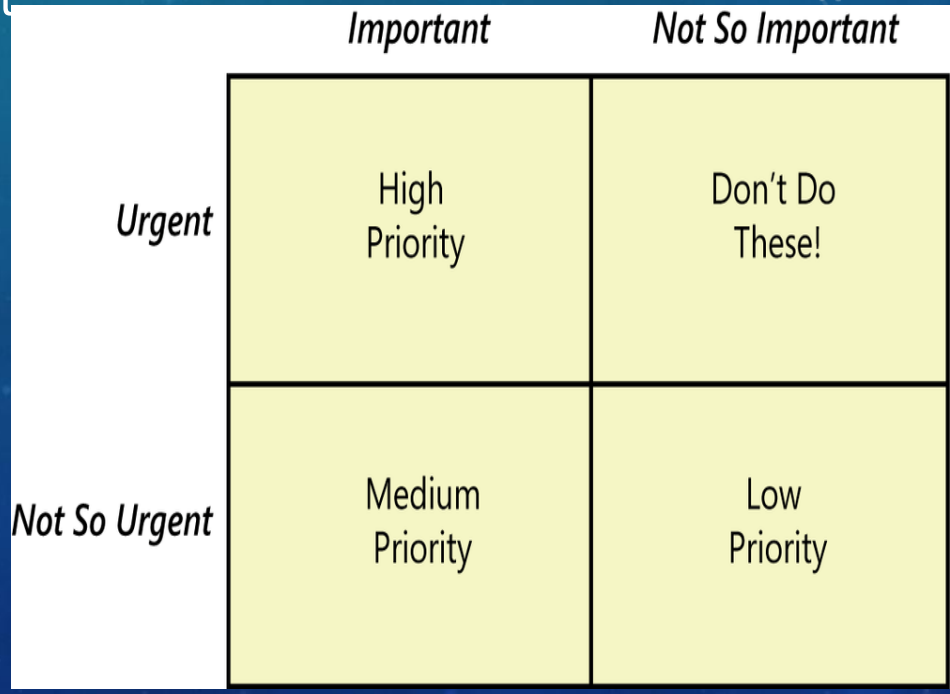
C, A, B, D, E

	A	B	C	D	E
A		<	^	<	<
B			^	<	<
C				<	<
D					<
E					

HOW TO PRIORITISE

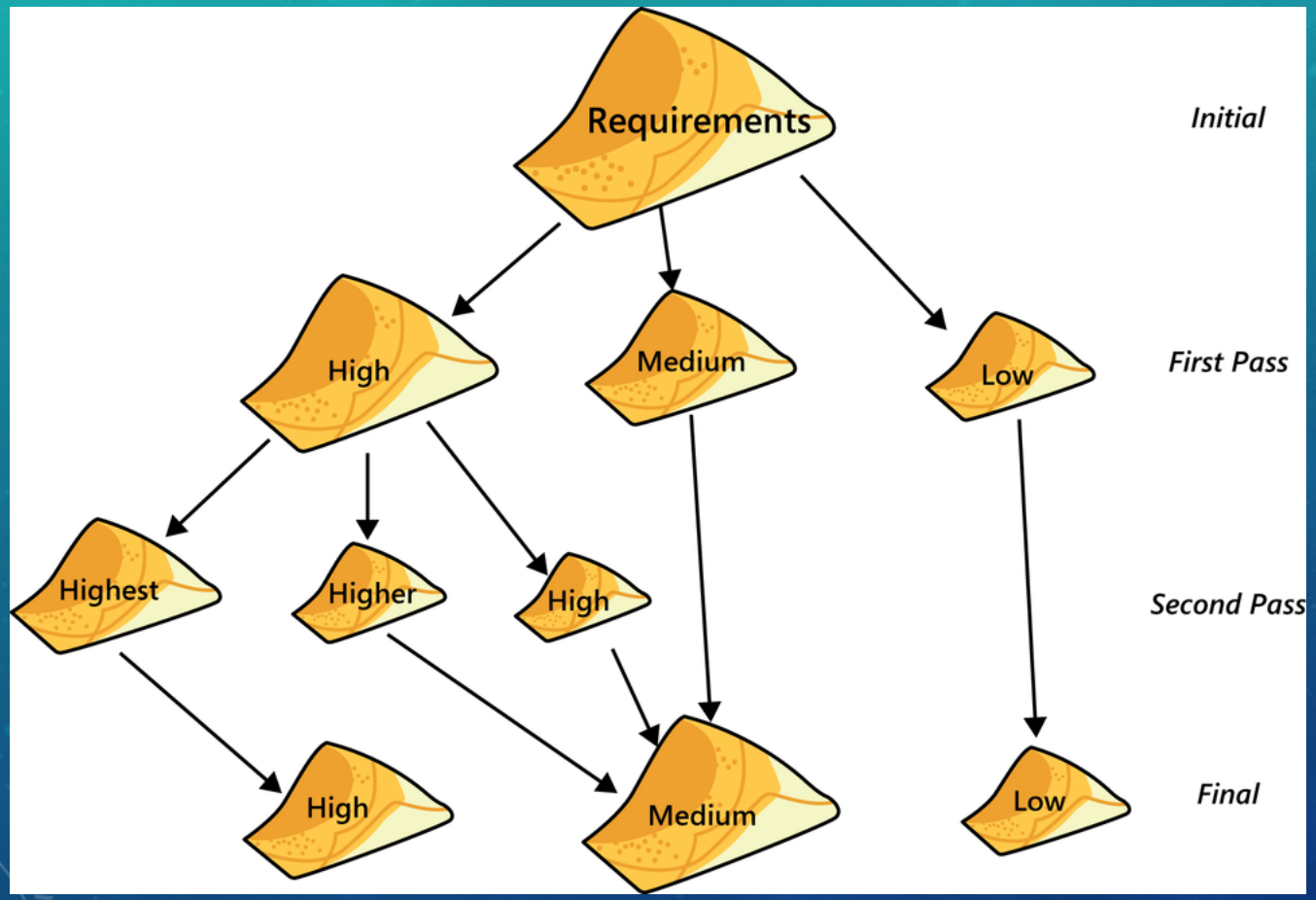
Three-level scale

- A common prioritisation approach groups requirements into three categories: high, medium, and low priority.
- One way to assess priority is to consider the two dimensions of importance and urgency
  - High-priority requirements are both important (customers need the capability) and urgent (customers need it in the next release).
  - Medium-priority requirements are important (customers need the capability) but not urgent (they can wait for a later release).
  - Low-priority requirements are neither important (customers can live without the capability if necessary) nor urgent (customers can wait, perhaps forever).
  - Requirements in the fourth quadrant appear to be urgent, perhaps for political reasons, but not important in terms of the system's functionality. So no need waste your time working on these.



HOW TO PRIORITISE

- Then, Repeat the process in the following way:



HOW TO PRIORITISE

MSCW model

- The four capitalized letters in the MSCW prioritisation scheme stand for four possible priority classifications for the requirements in a set:
  - Must: The requirement must be satisfied for the solution to be considered a success.
  - Should: The requirement is important and should be included in the solution if possible, but it's not mandatory to success.
  - Could: It's a desirable capability, but one that could be deferred or eliminated. Implement it only if time and resources permit.
  - Won't: This indicates a requirement that will not be implemented at this time but could be included in a future release.
- The MSCW scheme changes the three-level scale of high, medium, and low into a four-level scale. It doesn't offer any rationale for making the decision about how to rate the priority of a given requirement compared to others.

HOW TO PRIORITISE

Value, cost, and risk model

- Value = benefit*weight + penalty*weight
- Cost = ∑cost(i)*weight(i)
- Risk = ∑risk(i)*weight(i)
- Priority = value/(cost + risk)

Prioritisation is an optimisation process. Any optimalisation process must have an objective. In mathematics and AI, the objective is described using the so-called objective function which has the objective as the dependent variable, something that can be manipulated to change the value of the objective as the decision variables, as well as something that cannot be changed as parameters. Standard optimisation process has the formular of minimising the objective described with the objective function under a set of constraints.
In the above,
Objective function -- Priority = (benefit*weight_b + penalty*weight_p) / (∑cost(i)*weight(i) + ∑risk(i)*weight(i))
Decision variables – benefit, penalty, cost(i), risk(i)
Parameters – all the weights
Constraints – cost < A and risk < B

PART FOUR: A TRUE STORY

1. USER FIRST

Karen is a business analyst on a project to implement a new online catalogue for the company's customer service representatives (CSRs). The draft SRS is going through review when the marketing manager says he wants to add a "Like this product" feature. Karen's first instinct is to push back, there is already concern about meeting schedules with the current requirements set. But then she realised that may be that is a smart feature to add, because CSRs can promote the most-liked products with other customers. Before she elicits and documents the functional requirements for this feature, she needs an objective analysis about whether this feature should be added to the scope or not.

When she explained to the marketing manager the need to analyse this request feature, he responds, "well, soon the developers are going to be in there changing code anyway. How hard is it to add just a tiny feature?" Karen's analysis determines that the proposed feature lies outside the project's scope: It would not contribute the business objectives to reduce CSRs' average call time and it won't simple to implement. Karen needs to be able to clearly articulate why the feature is not in scope to the marketing manager, who doesn't have the business objectives readily in mind.

Qs: Who is the user? What is the business objective?

2. LUTON DSC

Disable Services Centre (DSC) provides its customers with services such as hiring and purchasing assistance equipment and rehabilitation equipment. DPS has a number of departments and branches in Luton and in other small towns near Luton.

Business objective: DSC wanted to replace its CRM system as the legacy system was not supported by the developer any more. DSC has to have a new one to manage the services provided to its customers.

User requirements elicitation: The heads of all departments and branches gathered together and described how business is running in their department/branches.

Diagrams on the next slide show how its Equipment Department operates.

Diagrams on Slides 44 to 45 show how we documented the requirements.

Equipment Department

Process

* walk in,
* out reach
* email/call

Info/data activity

Δ record: name, date, time, query, person(staff) ref to

? (pInfo DB)

Reception

Info desk

* contact person assigned (staff)
* equipment or other services
* stock check
* "call back"

Δ record ① further personal info (PI DB)
② contact person's name (staff) (PG DB)
③ call back details

Δ access to stock DB

Δ access to other services DB

further assessment 4 equipment + mobility aids

* making decisions on either refer to other agency or sale, hire, sign-posting

Δ access to PG DB to
① record the decision.
② assessment report

Δ record ① progress/decision.
② assessment report

Equipment assessment/ trial / selection

* make decision on hiring, purchase or special order

Hiring

* check resources (in stock ? services ?)
* demo/instruction
* draft/sign agreement
* deposit taking
* remind expiration day (about send)
* either
  - returned: inspection
    deposit refund
  - or
  - extension: availability
    payment

Δ access to stock DB
Δ record agreement details
Δ financing info
Δ maintenance info
Δ manager informed

OR:

purchase

* check stock
* decision on vat
* invoice
* payment
* "return policy" (inspection ?)

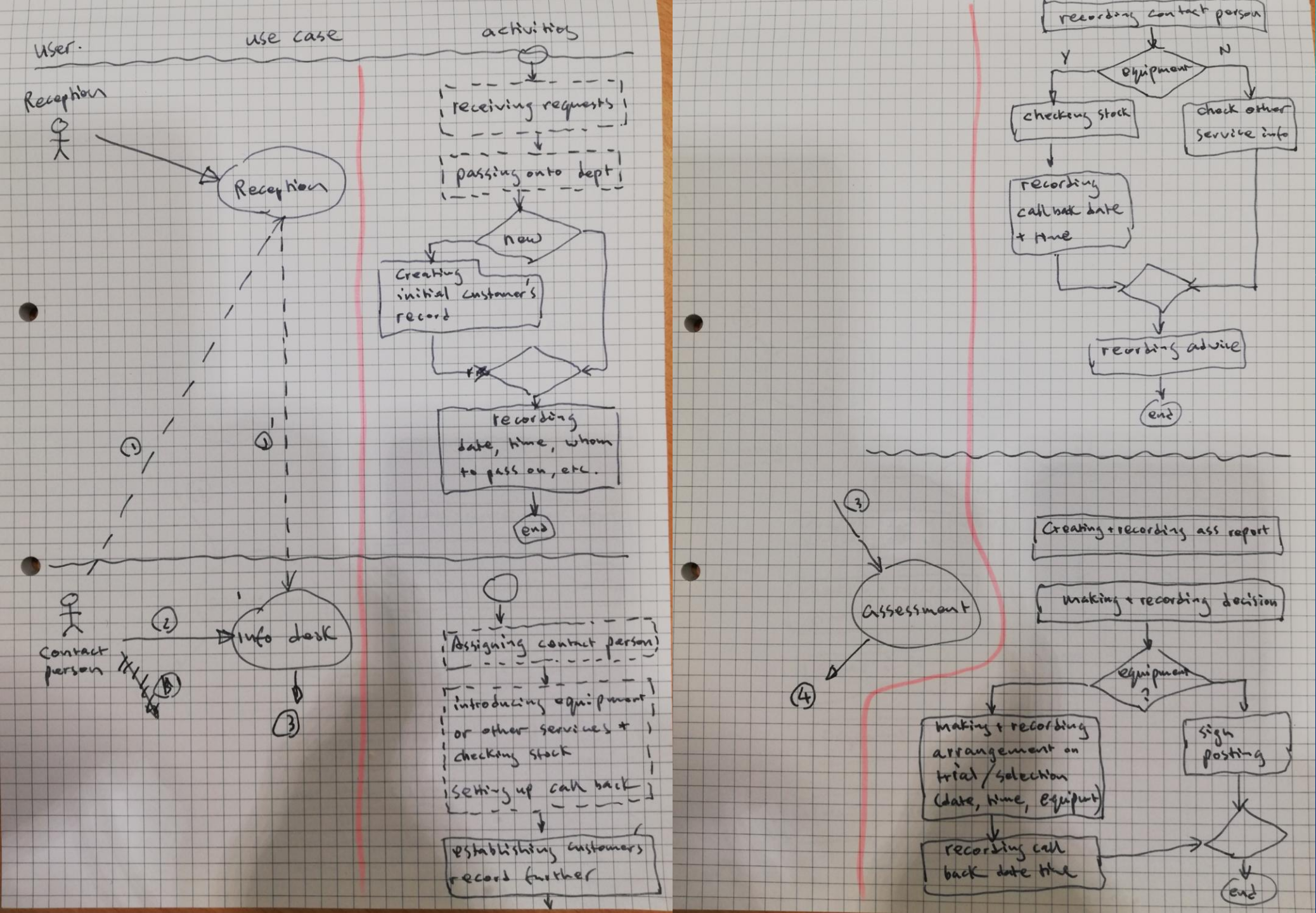
Δ access to stock DB
Δ access to VAT info
Δ financing
Δ manager informed
Δ record end-user info

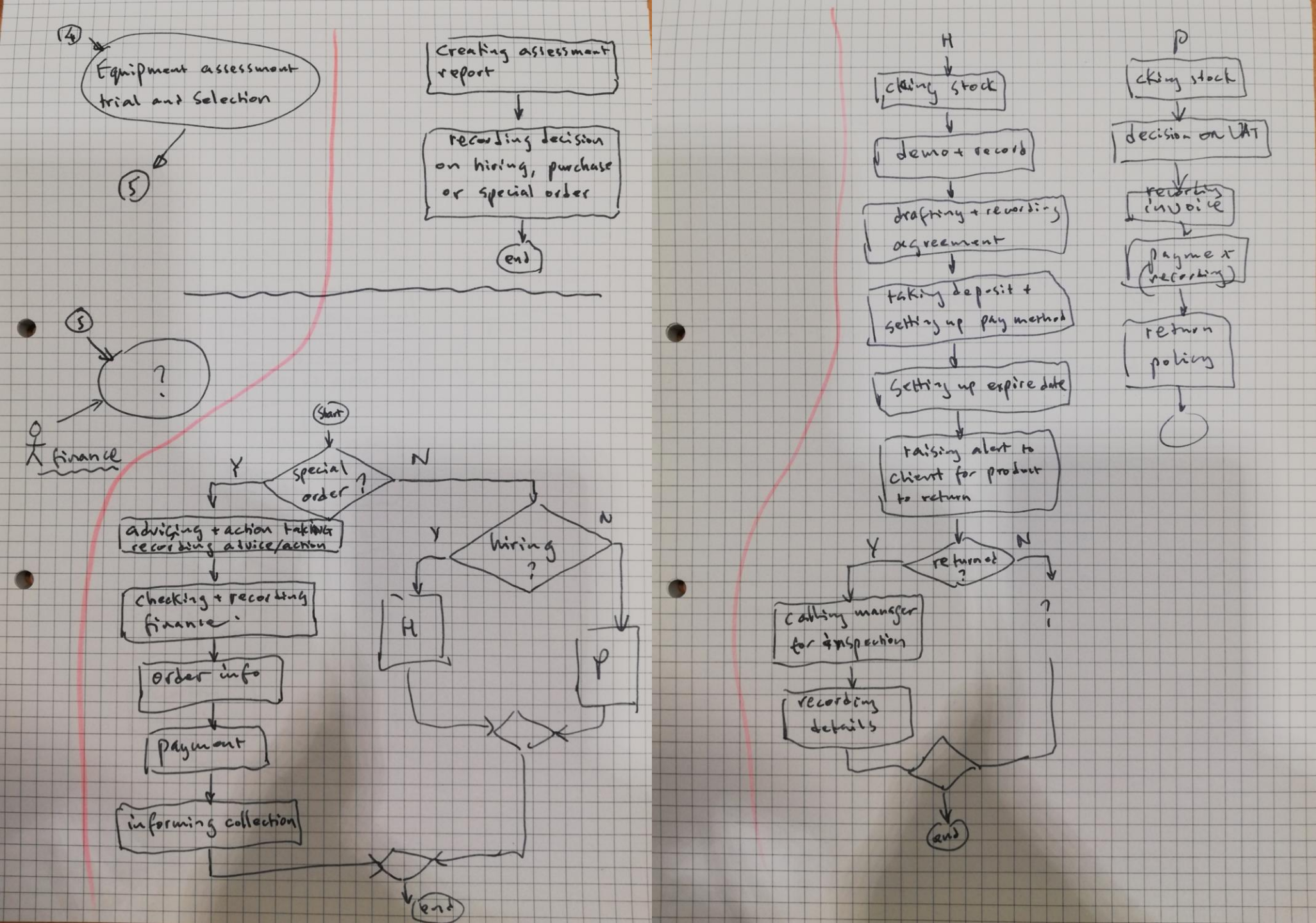
OR

special ordered

* advice on either sample models or suppliers
* place order
* payment
* inform client to collect

Δ sample/supplier info
Δ financing
Δ order info (record)
name, date, address, equipment, etc.
Δ stock (to deal with not paied not collected)


User.
use case
activities
Reception
Reception
receiving requests
passing onto dept
new
Creating initial customers record
recording date, time, whom to pass on, etc.
end
1
4
2
Contact person
Info desk
3
assigning contact person
introducing equipment or other services + checking stock
setting up call back
establishing customers record further
recording contact person
Y
equipment
N
checking stock
check other service info
recording call back date + time
recording advice
end
3
Assessment
4
Creating + recording ass report
making + recording decision
equipment ?
making + recording arrangement on trial / selection (date, time, equipt)
recording call back date time
sign posting
end


4
Equipment assessment trial and selection
5
Creating assessment report
recording decision on hiring, purchase or special order
end
5
?
finance
start
special order ?
Y
N
adviging + action taking recording advice/action
checking + recording finance
order info
payment
informing collection
hiring ?
Y
N
H
P
end
H
cking stock
demo + record
drafting + recording agreement
taking deposit + setting up pay method
setting up expire date
raising alert to client for product to return
returned ?
Y
N
calling manager for inspection
recording details
?
end
P
cking stock
decision on UNIT
recording invoice
payment recording
return policy

2. LUTON DSC

Interruption: DSC manager stepped in to override the user requirements collected from the meetings, leading to internal disagreements withing the company, and added in more and more requirements which is beyond the scope of that project

Disaster: DSC do not have personnel who is able to transfer data from the old system to the new one and the software company that developed the old system refused to help.