
MÃ GIẢM GIÁ
TIKI.VN
TIKI.VN

Describing Process Specification and Structured Decisions

Group 6

Members

Nguyen Viet Sang

Hoang Tran Minh Tai

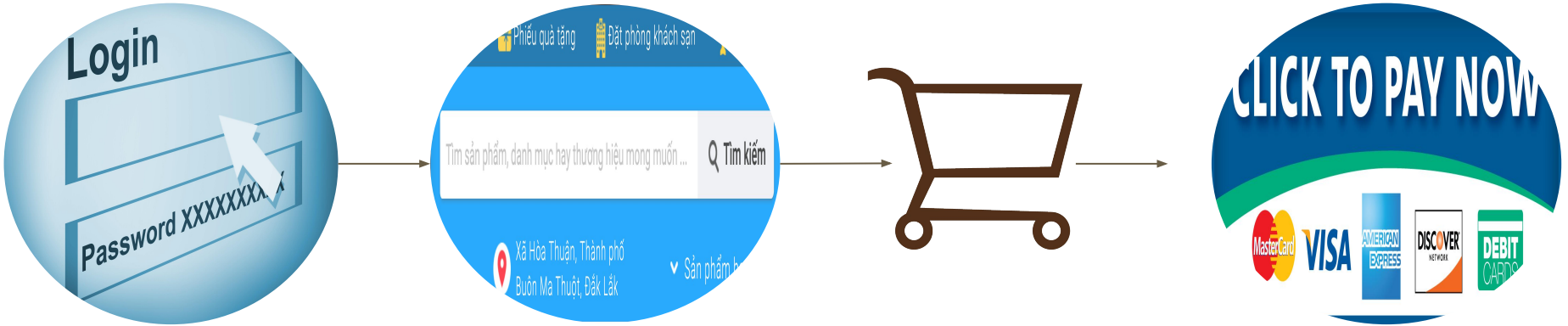
Tran Ngoc Quy

Nguyen Quang Sang

Overview

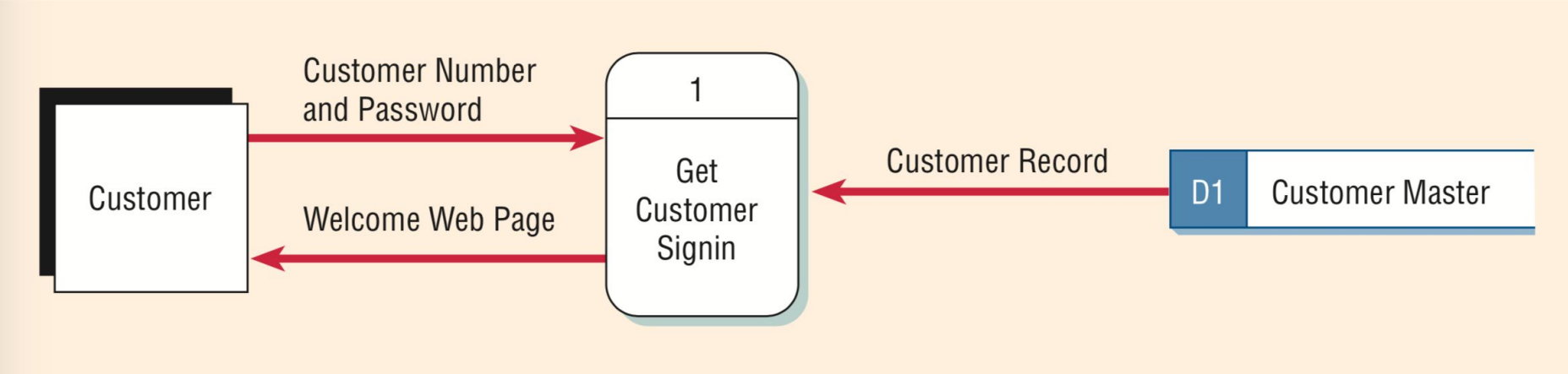
Have you ever bought an item on TIKI?

Overview



Procedure

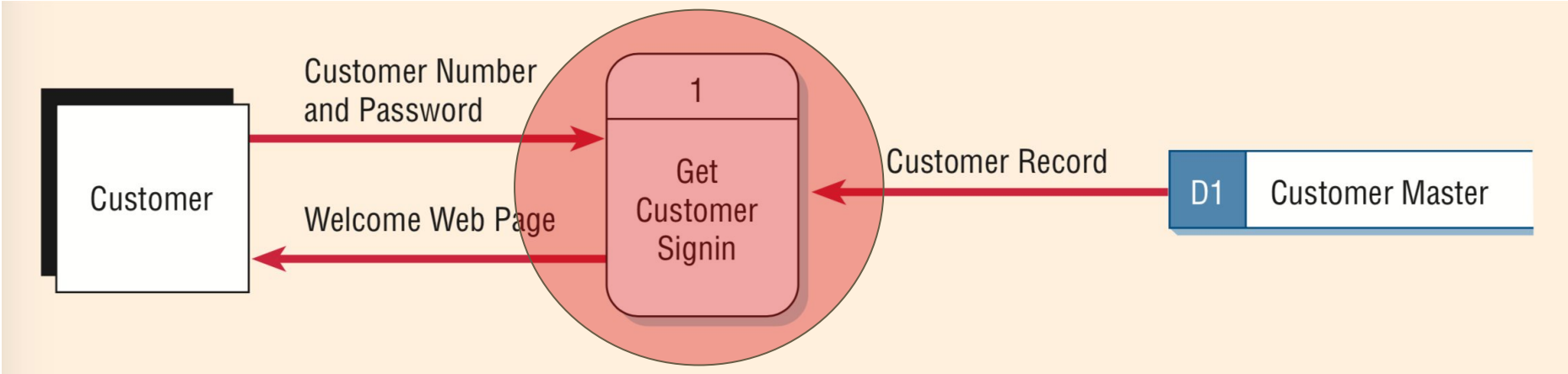
Overview



Data Flow Diagram for Login

Overview

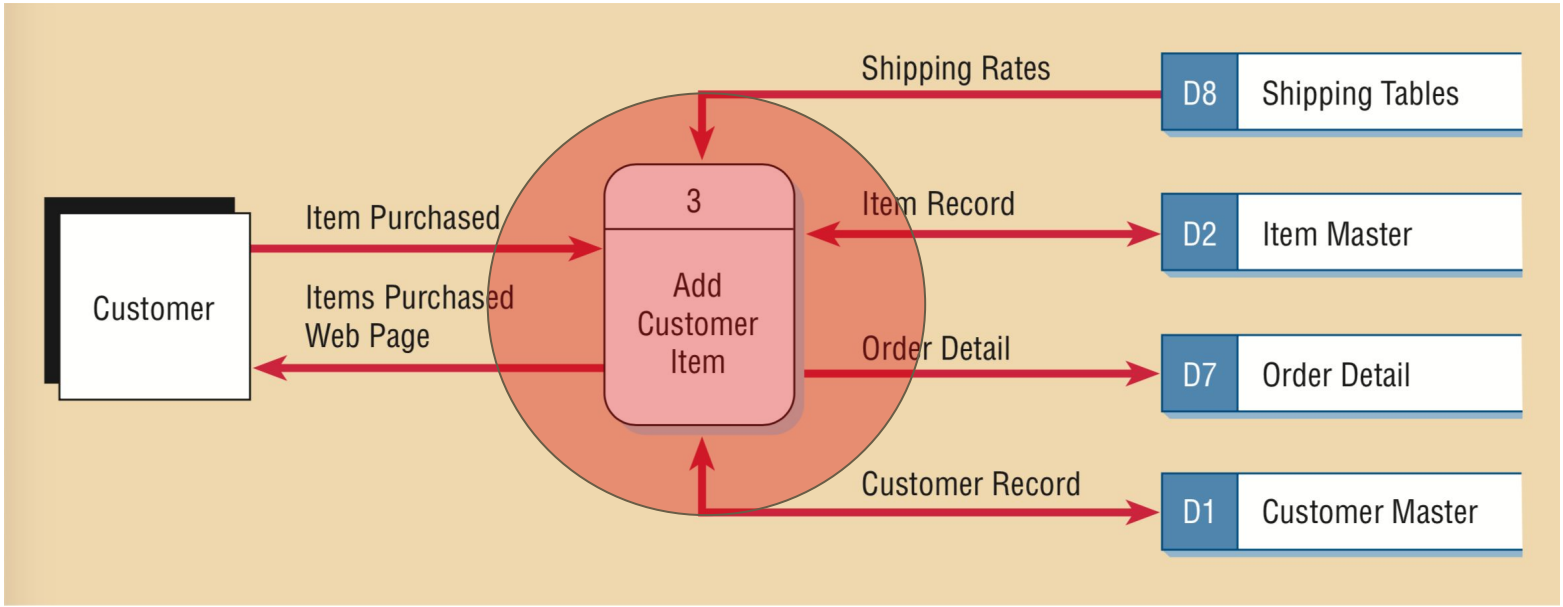
What does process actually do?



Data Flow Diagram for Login

Overview

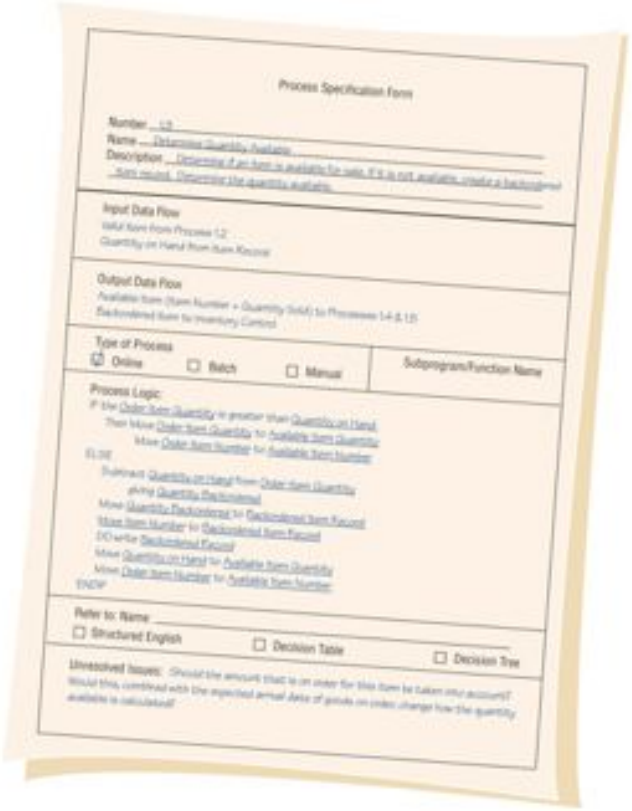
What does process actually do?



Data Flow Diagram for Purchase

Overview

Need **Process Specification**

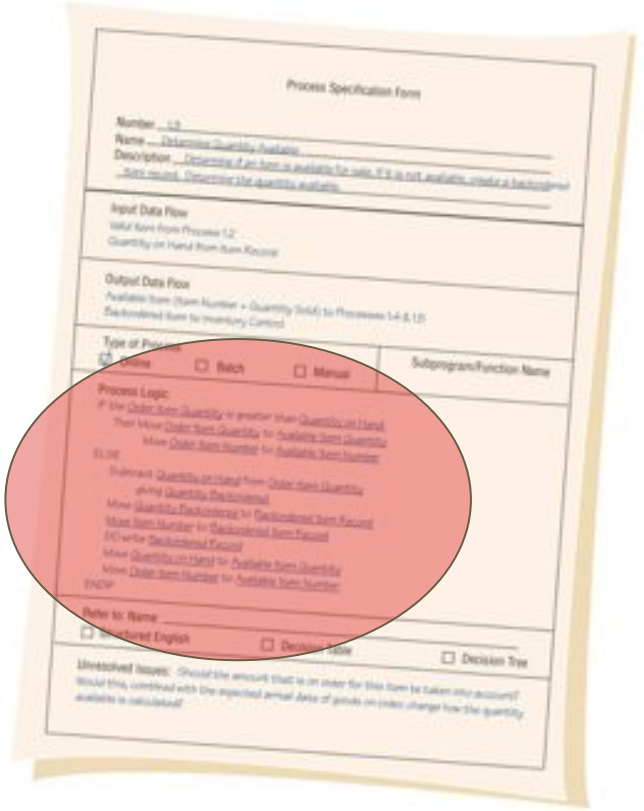


Overview

Need **Process Specification**

support
to clarify

Structured Decisions



Content

- Process specifications
- Structured English
- Decision tables
- Decision trees
- Horizontal balancing

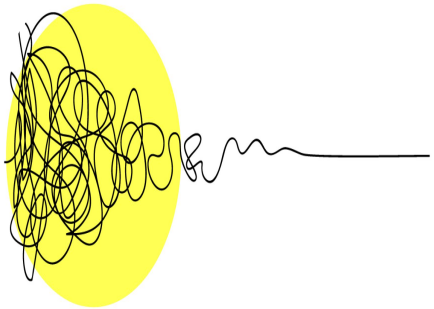
Content

- Process specifications
- Structured English
- Decision tables
- Decision trees
- Horizontal balancing

Process specifications

- Sometimes called **minispecs**
- Create for
  - processes on a data flow diagram
  - class methods in object-oriented design
  - steps in use case

Process specifications - Goals



Reduce process ambiguity

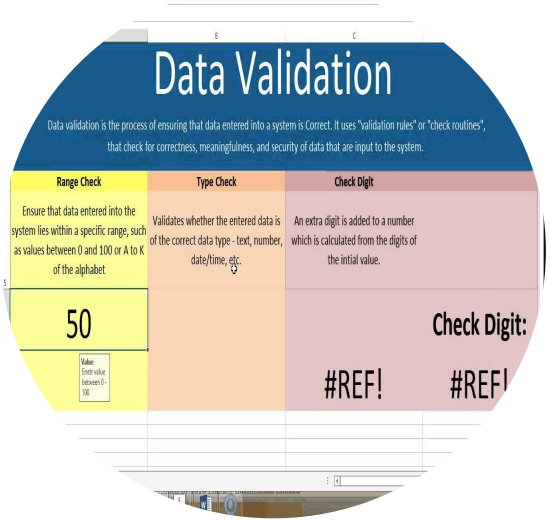
Precise description
of what is accomplished

↓

Validate the system design

Process specifications - Not create for processes that...

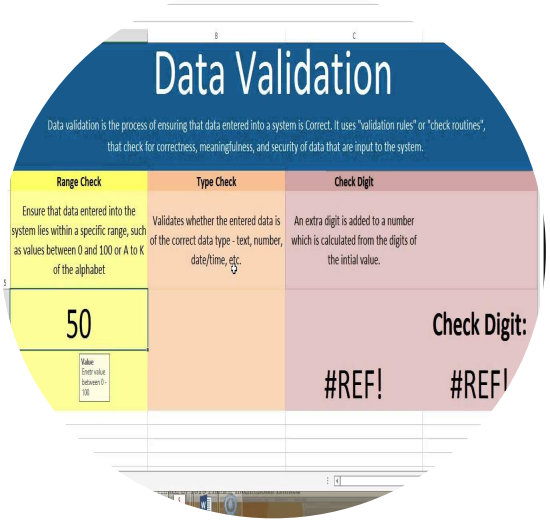
Physical input/output



Simple data validation

Prewritten code

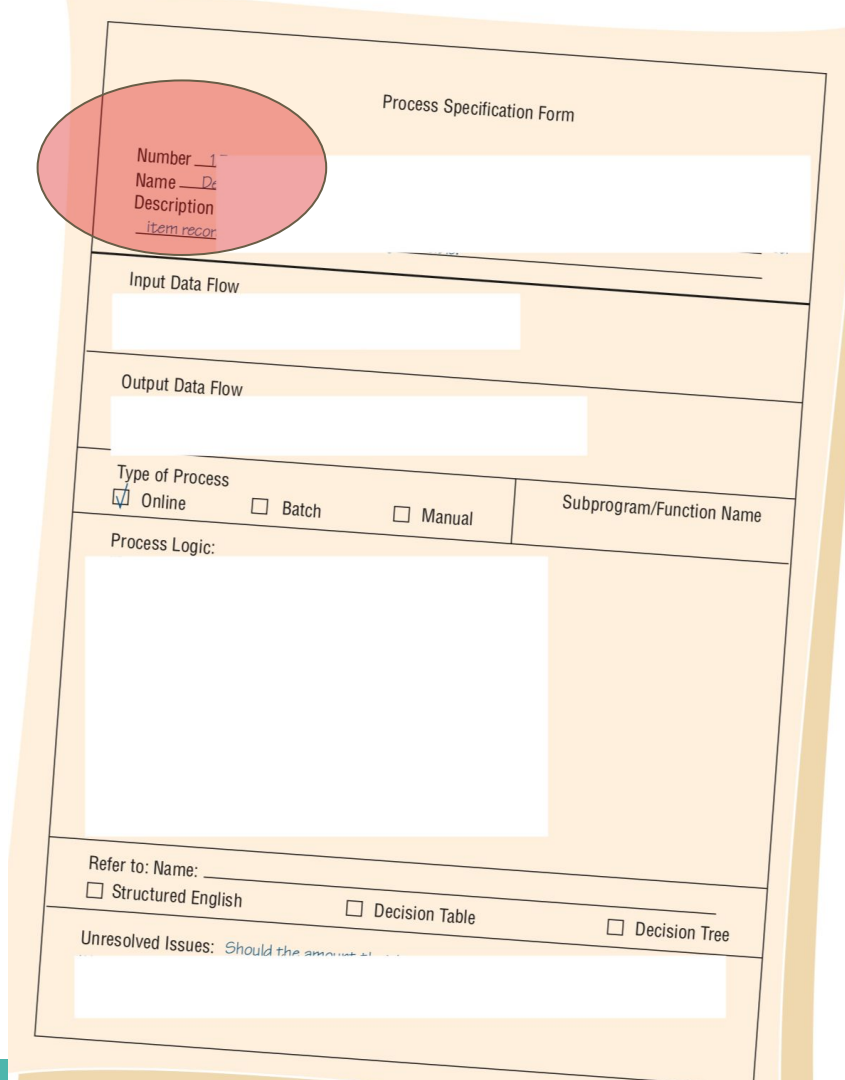
Process specifications - Not create for processes that...



Simple or computer code already exists!

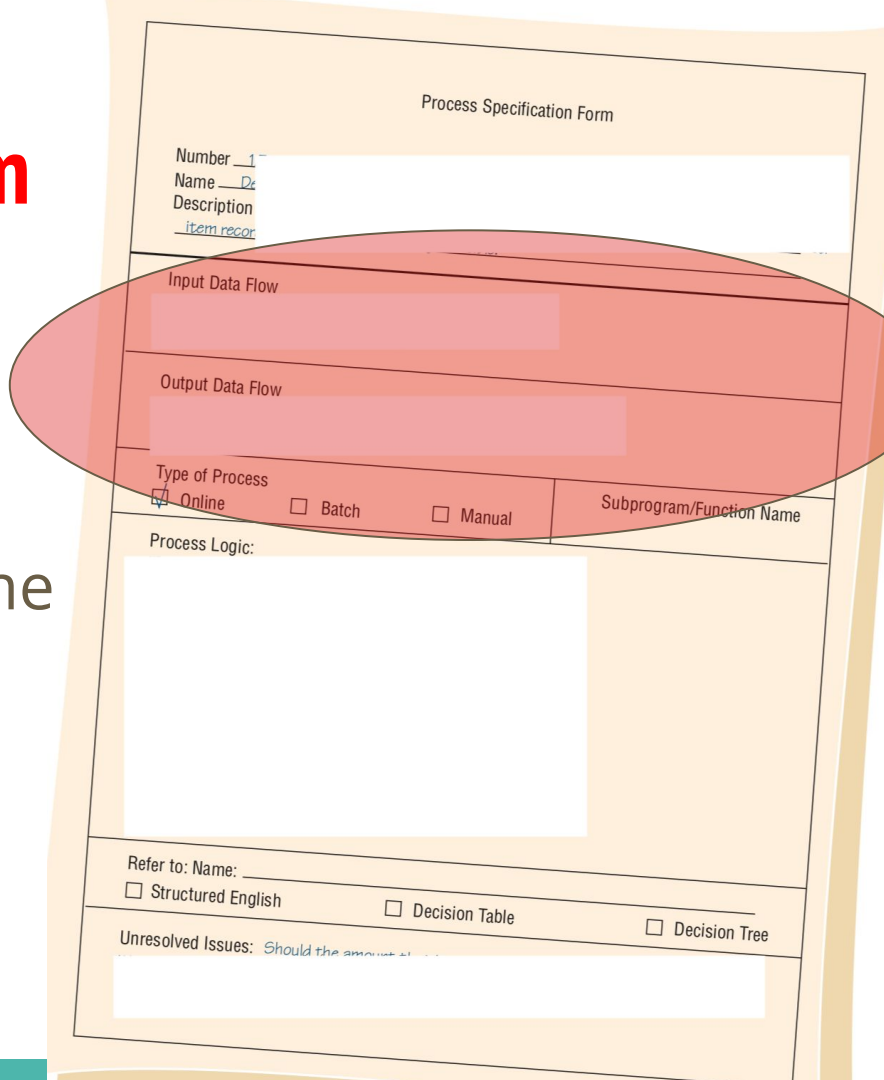
Process specifications - Form

- Number
- Name
- Description: what the process accomplishes



Process specifications - Form

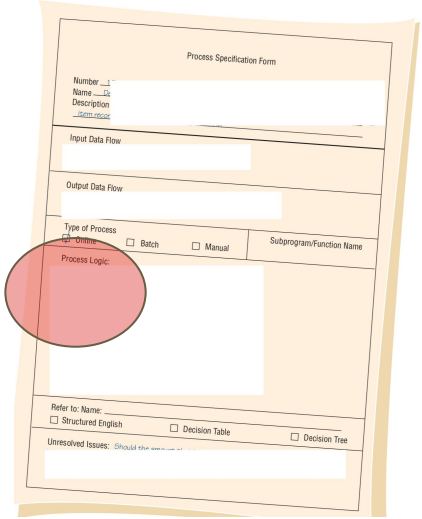
- Input Data Flow
- Output Data Flow
- Type of Process
- Subprogram/Function Name
  E.g: library name



Process specifications - Form

- Process Logic: states policy and **business rules** (conditions, formulas...)

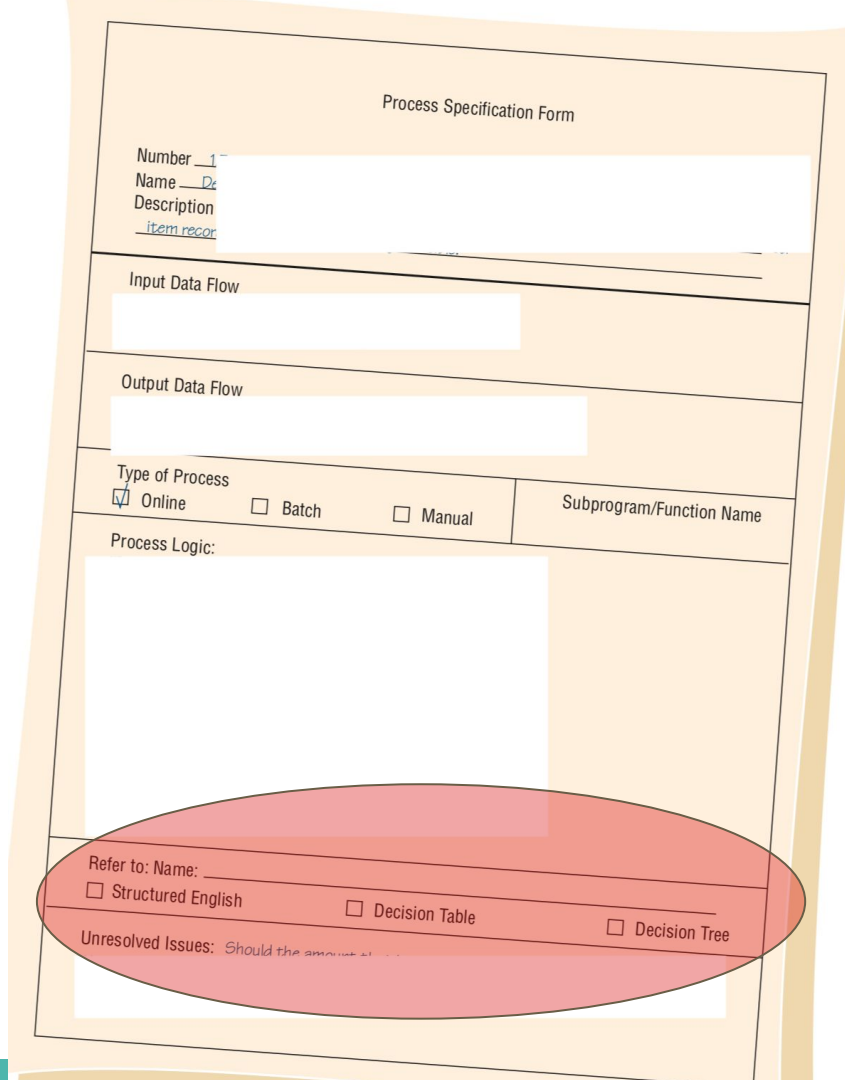
Process specifications - Form



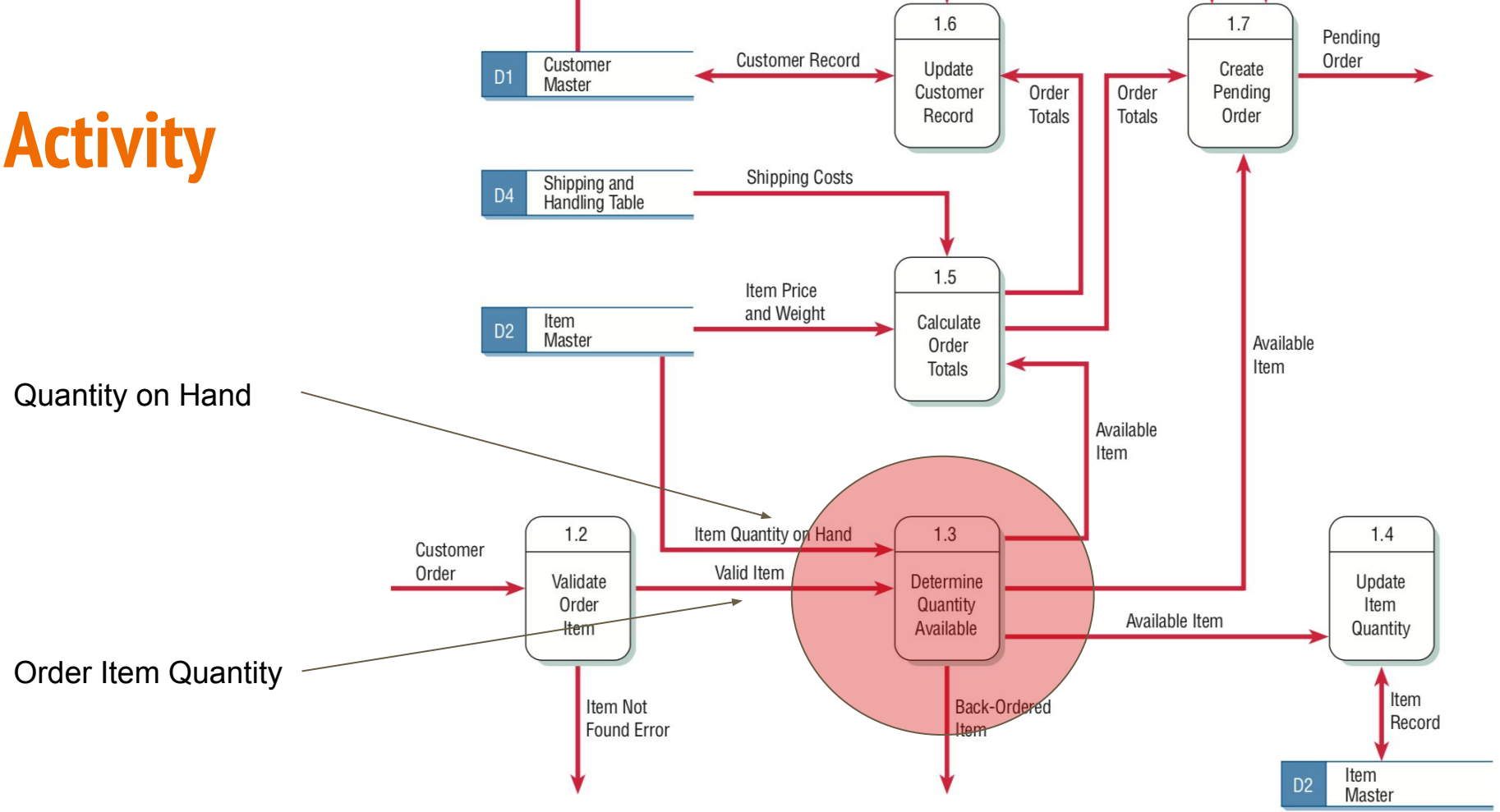
- Common Business rules:
  - Definitions of business terms
  - Business conditions and actions
  - Data integrity constraints
  - Mathematical and functional derivations
  - Logical inferences
  - Processing sequences
  - Relationships among facts about the business

Process specifications - Form

- Logic method reference: If not enough room
- Unresolved Issues

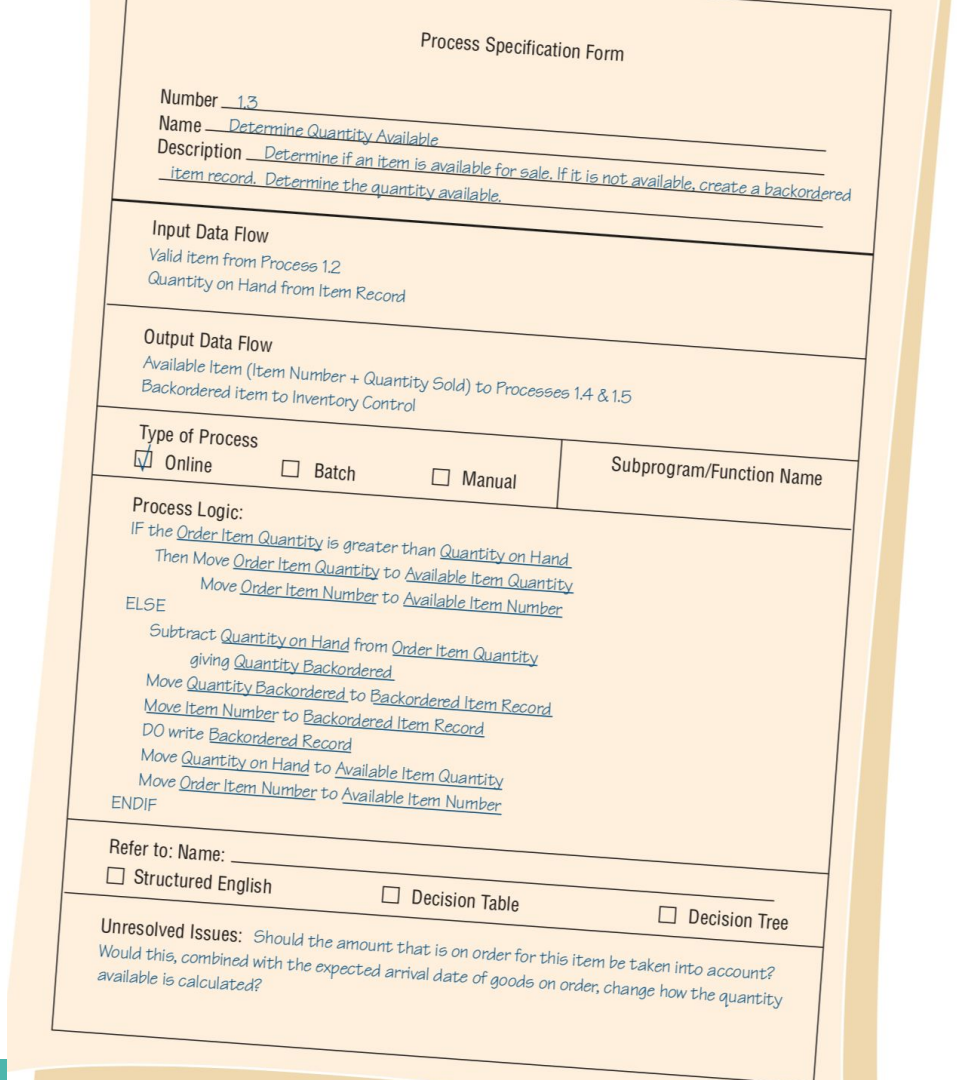


Activity



Fill the Process Specification for this process

Solution



Process Specification Form

Number 1.3
Name Determine Quantity Available
Description Determine if an item is available for sale. If it is not available, create a backordered item record. Determine the quantity available.

Input Data Flow
Valid item from Process 1.2
Quantity on Hand from Item Record

Output Data Flow
Available Item (Item Number + Quantity Sold) to Processes 1.4 & 1.5
Backordered item to Inventory Control

Type of Process
☑ Online ☐ Batch ☐ Manual

Subprogram/Function Name

Process Logic:
IF the Order Item Quantity is greater than Quantity on Hand
Then Move Order Item Quantity to Available Item Quantity
Move Order Item Number to Available Item Number
ELSE
Subtract Quantity on Hand from Order Item Quantity
giving Quantity Backordered
Move Quantity Backordered to Backordered Item Record
Move Item Number to Backordered Item Record
DO write Backordered Record
Move Quantity on Hand to Available Item Quantity
Move Order Item Number to Available Item Number
ENDIF

Refer to: Name:
☐ Structured English ☐ Decision Table ☐ Decision Tree

Unresolved Issues: Should the amount that is on order for this item be taken into account? Would this, combined with the expected arrival date of goods on order, change how the quantity available is calculated?

Content

- Process specifications
- Structured English
- Decision tables
- Decision trees
- Horizontal balancing

What is Structured English?

- The use of English Language with syntax of programming structure
- Be like limited-form of "pseudocode"
- Used for:
  - Communicate Computer Program - Non-tech users
  - Give aims to get benefit of both Programming logic - Natural Language
  - Logic involves: Formulas, Iterations, ...
- 4 Structured English Types:
  - Sequential Structure
  - Decision Structure
  - Case Structure
  - Iteration

Writing Structured English

- All logic should be expressed in 1 of 4 types
- Statements should be clear and unambiguous
- Indent blocks of statements
- One line or indent next line
- CAPITALIZE keywords
- CAPITALIZE name of block statements, finish with END
- Underline words defined in data dictionary
- Comment line with *

Writing Structured English

- 4 Structured English Types
- Keywords

Structured English Type	Example
Sequential Structure A block of instructions in which no branching occurs	Action #1 Action #2 Action #3
Decision Structure Only IF a condition is true, complete the following statements; otherwise, jump to the ELSE	IF Condition A is True THEN implement Action A ELSE implement Action B ENDIF
Case Structure A special type of decision structure in which the cases are mutually exclusive (if one occurs, the others cannot)	IF Case #1 implement Action #1 ELSE IF Case #2 Implement Action #2 ELSE IF Case #3 Implement Action #3 ELSE IF Case #4 Implement Action #4 ELSE print error ENDIF
Iteration Blocks of statements that are repeated until done	DO WHILE there are customers. Action #1 ENDDO

Example

```
IF customer has a Bank Account THEN
  IF Customer has no dues from previous account THEN
    Allow loan facility
  ELSE
    IF Management Approval is obtained THEN
      Allow loan facility
    ELSE
      Reject
    ENDIF
  ENDIF
ELSE
  Reject
ENDIF
EXIT
```

Data Dictionary?

- Is a structured repository of data elements in system.
- Stores description of all DFD data elements
- Such as: data flows, data stores, the processes

Example:
GET Order Record
GET Customer Record
Move Order Number to shipping statement
Move Order Date to Shipping Statement
Move Customer Number to Shipping Statement
...

Advantages of Structured English

- Clarifying the logic and relationships found in human languages
- Effective communication tool
- It can be taught to and understood by users in the organization

Content

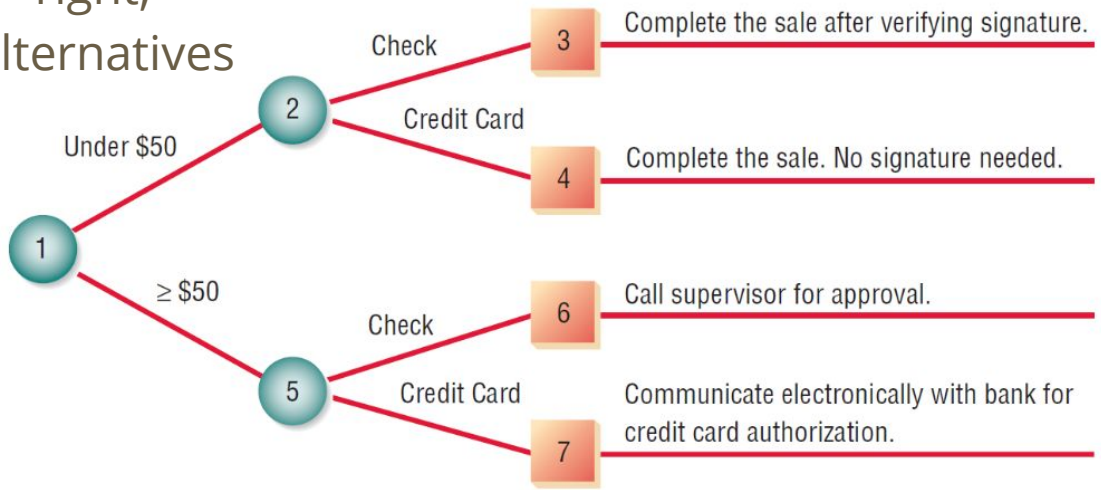
- Process specifications
- Structured English
- Decision trees
- Decision tables
- Horizontal balancing

Decision Trees

- Used when complex branching occurs in a structured decision process
- Keep a string of decisions in a particular sequence

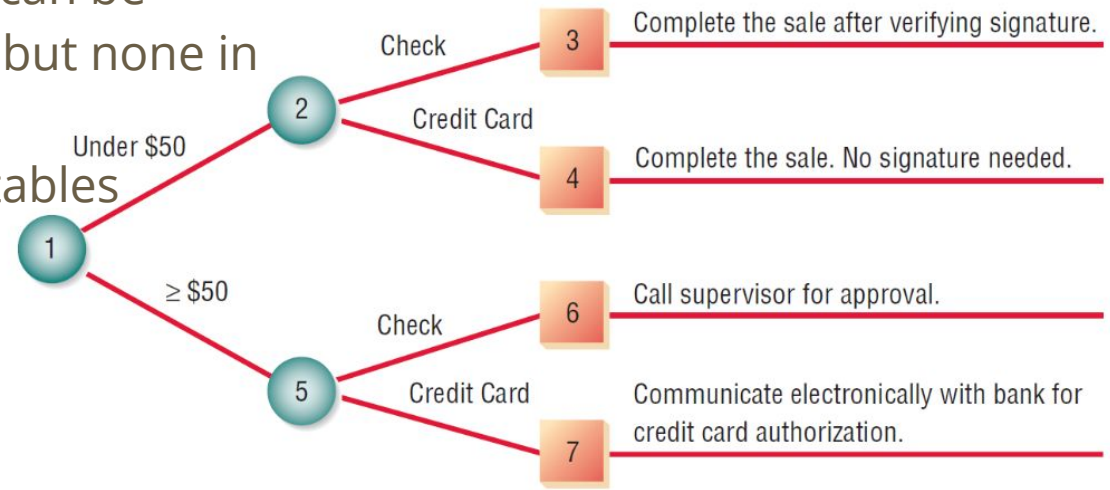
Drawing Decision Trees

- Identify all conditions, actions, order, timings
- Building Tree from left -> right, make sure all possible alternatives moving to the right



Decision Trees Advantages

- The order of checking conditions is immediately noticeable
- Conditions and actions can be available in this branch but none in the other
- Compared to Decision tables more readily understood



Content

- Process specifications
- Structured English
- Decision trees
- Decision tables
- Horizontal balancing

Decision table

- It is a graphical method for representing process logic.
- It provide a way to examine, describe, and document decisions using a table.
- They are used to:
  - Describe the conditions.
  - Identify possible decision alternatives.
  - Indicate actions should be performed.
  - Describe actions.

Figure 9.9 The standard format used for presenting a decision table.

Conditions and Actions	Rules
Conditions	Condition Alternatives
Actions	Action Entries

Decision Table Example

Rule 3 says:

If N (the total sale is NOT under $50)
AND
If Y (the customer paid by check and had two forms of ID)
AND
If N (the customer did not use a credit card)
THEN
DO X (call the supervisor for approval)

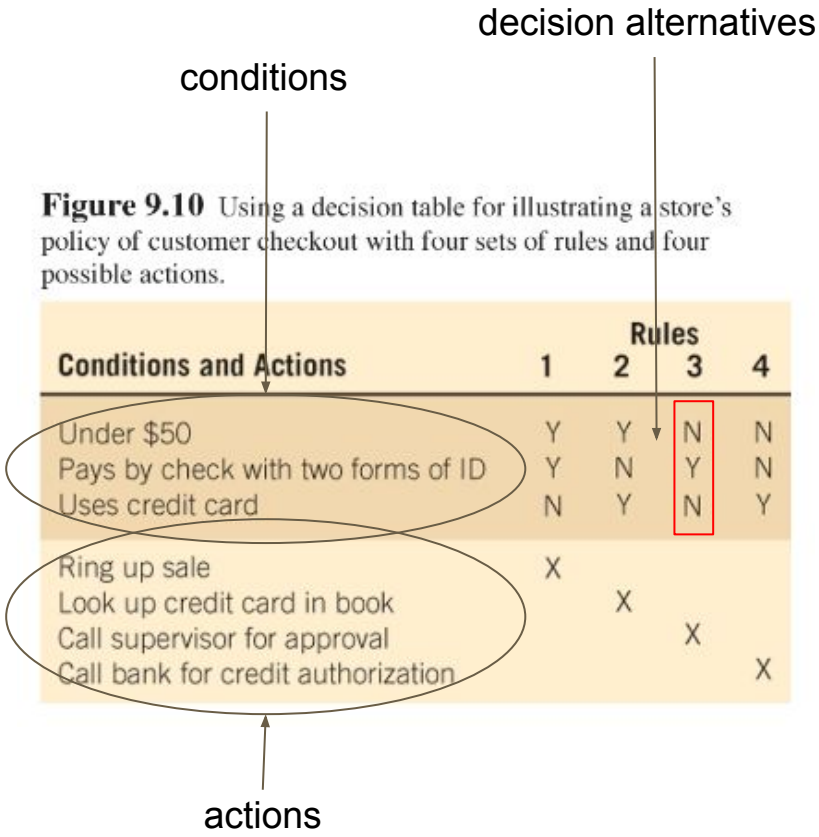


Figure 9.10 Using a decision table for illustrating a store's policy of customer checkout with four sets of rules and four possible actions.

Conditions and Actions	Rules 1	2	3	4
Under $50	Y	Y	N	N
Pays by check with two forms of ID	Y	N	Y	N
Uses credit card	N	Y	N	Y
Ring up sale	X			
Look up credit card in book		X		
Call supervisor for approval			X	
Call bank for credit authorization				X

Developing Decision Tables

To build decision tables, an analyst needs to determine:

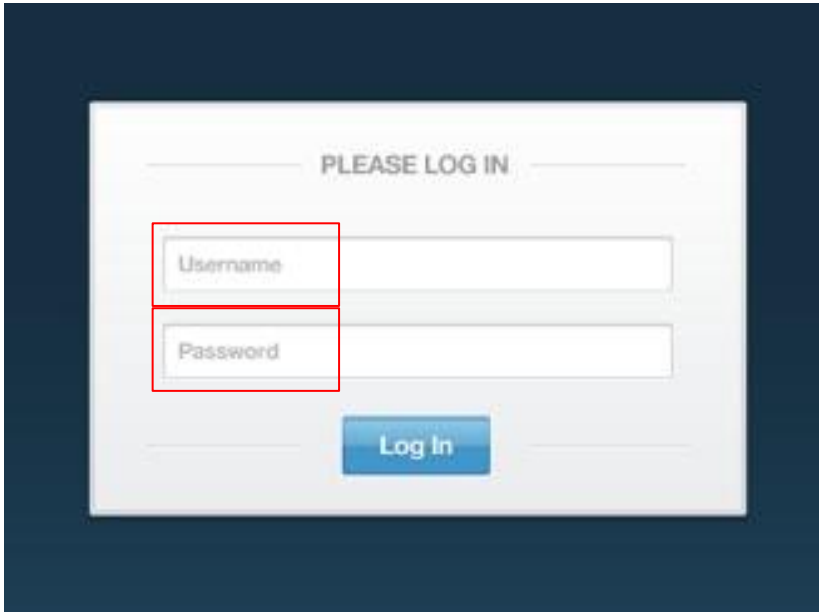
- The maximum size of the table
- Eliminate any impossible situations, inconsistencies, or redundancies
- Simplify the table as much as possible

Developing Decision Tables - Login function

Developing Decision Tables - Login function

Step 1: Determine conditions that affect the decision

- Combine rows that overlap, such as conditions that are mutually exclusive.
- The number of conditions becomes The number of rows in the top half of the decision table.



Developing Decision Tables - Login function

Conditions and Actions	Rules						
Username							
Password							

Developing Decision Tables - Login function

Step 2: Determine the number of possible actions that can be taken.

That number becomes the number of rows in the lower half of the decision table.

- Show error: Please enter username
- Show error: Please enter a valid username
- Show error: Login failed
- Show error: Please enter password
- Show home page

Developing Decision Tables - Login function

Conditions and Actions	Rules						
Username							
Password							
Show error: Please enter username							
Show error: Please enter a valid username							
Show error: Login failed							
Show error: Please enter password							
Show home page							

Developing Decision Tables - Login function

Step 3: Determine the number of condition alternatives for each condition.

- Username:
  - Blank
  - Invalid
  - Valid
- Password:
  - Blank
  - Invalid
  - Valid

Developing Decision Tables - Login function

Step 4: Calculate the maximum number of columns in the decision table by multiplying the number of alternatives for each condition.

Condition 1: x 3 alternatives

Condition 2: x 3 alternatives

9 possibilities

Developing Decision Tables - Login function

Conditions and Actions	Rules								
	1	2	3	4	5	6	7	8	9
Username									
Password									
Show error: Please enter username									
Show error: Please enter a valid username									
Show error: Login failed									
Show error: Please enter password									
Show home page									

Developing Decision Tables - Login function

Step 5: Fill in the condition alternatives.

(Blank - B, Invalid - I, Valid - V)

Condition 1: B B B I I I V V V

Condition 2: B I V B I V B I V

Step 6: Complete the table by inserting an X where rules suggest certain actions.

Developing Decision Tables - Login function

Conditions and Actions	Rules								
	1	2	3	4	5	6	7	8	9
Username	B	B	B	I	I	I	V	V	V
Password	B	I	V	B	I	V	B	I	V
Show error: Please enter username	X	X	X						
Show error: Please enter a valid username				X		X			
Show error: Login failed					X			X	
Show error: Please enter password							X		
Show home page									X

Developing Decision Tables - Login function

Step 7: Combine rules where it is apparent that an alternative does not make a difference in the outcome.

Username:	B	B	B		B
Password:	B	I	V	→	—
Enter username:	X	X	X		X

Developing Decision Tables - Login function

Step 8: Check the table for any impossible situations, contradictions, and redundancies. They are discussed in more detail later.

Step 9: Rearrange the conditions and actions (or even rules) if it makes the decision table more understandable.

Developing Decision Tables - Login function

Conditions and Actions	Rules						
	1	2	3	4	5	6	7
User name	B	I	I	I	V	V	V
Password	_	B	I	V	B	I	V
Show error: Please enter username	X						
Show error: Please enter a valid username		X		X			
Show error: Login failed			X			X	
Show error: Please enter password					X		
Show home page							X

Checking for Completeness and Accuracy

- Decision tables help analysts ensure completeness and accuracy.
- Four main problems that can occur in developing decision tables:
  - Incompleteness.
  - Impossible situations.
  - Contradictions.
  - Redundancy.

Checking for Completeness and Accuracy

Conditions and Actions	Rules 1	2	3	4
Salary > $50,000/year	Y	Y	N	N
Salary < $2,000/month	Y	N	Y	N
Action 1				
Action 2				

Rule 1 is not feasible, because a person cannot earn greater than $50,000 per year and less than $2,000 per month at the same time.

This is an impossible situation.

Conditions and Actions	Rules 1	2	3	4	5	6	7
Condition 1	Y	Y	Y	Y	Y	N	N
Condition 2	Y	Y	Y	N	N	Y	N
Condition 3	—	N	—	—	—	N	Y
Action 1	X			X	X		
Action 2			X			X	
Action 3		X					X

Contradiction

Redundancy

Contradictions often occur of dashes [--] are incorrectly inserted into the table. Redundancy occurs when identical sets of alternatives require the exact same action. The analyst has to determine what is correct and then resolve the contradiction or redundancy.

More Advanced Decision Tables

- Decision tables can become very big because they grow rapidly as the number of conditions and alternatives increases.
- In order to avoid a decision table to grow rapidly, we can use Extended Entries or ELSE rule.
  - Extended entries: to reduce the possibility of redundancy and contradiction.

C1: Did not order	YNNN
C2: Ordered once	NYNN
C3: Ordered twice	NNYN
C4: Ordered more than twice	NNNY

C1: Number of times customer ordered: 0 1 2 >2

More Advanced Decision Tables

2. ELSE Rule: to eliminate repetitive rules requiring the exact same action.

Conditions and Actions	Rules 1	2	3	4	ELSE
Cost of the item A cost < $10 B $10 ≤ cost ≤ $50 C cost > $50	–	A	B	C	
Order quantity D quantity < 50 E 50 ≤ quantity ≤ 100 F quantity > 100	D	E	E	F	
Order immediately			X		
Wait until regular order is placed	X	X			
Send to purchasing for bid				X	
Check with supervisor					X

Content

- Process specifications
- Structured English
- Decision tables
- Decision trees
- Horizontal balancing

Content

- Process specifications
- Structured English
- Decision tables
- Decision trees
- Horizontal balancing

Horizontal Balancing

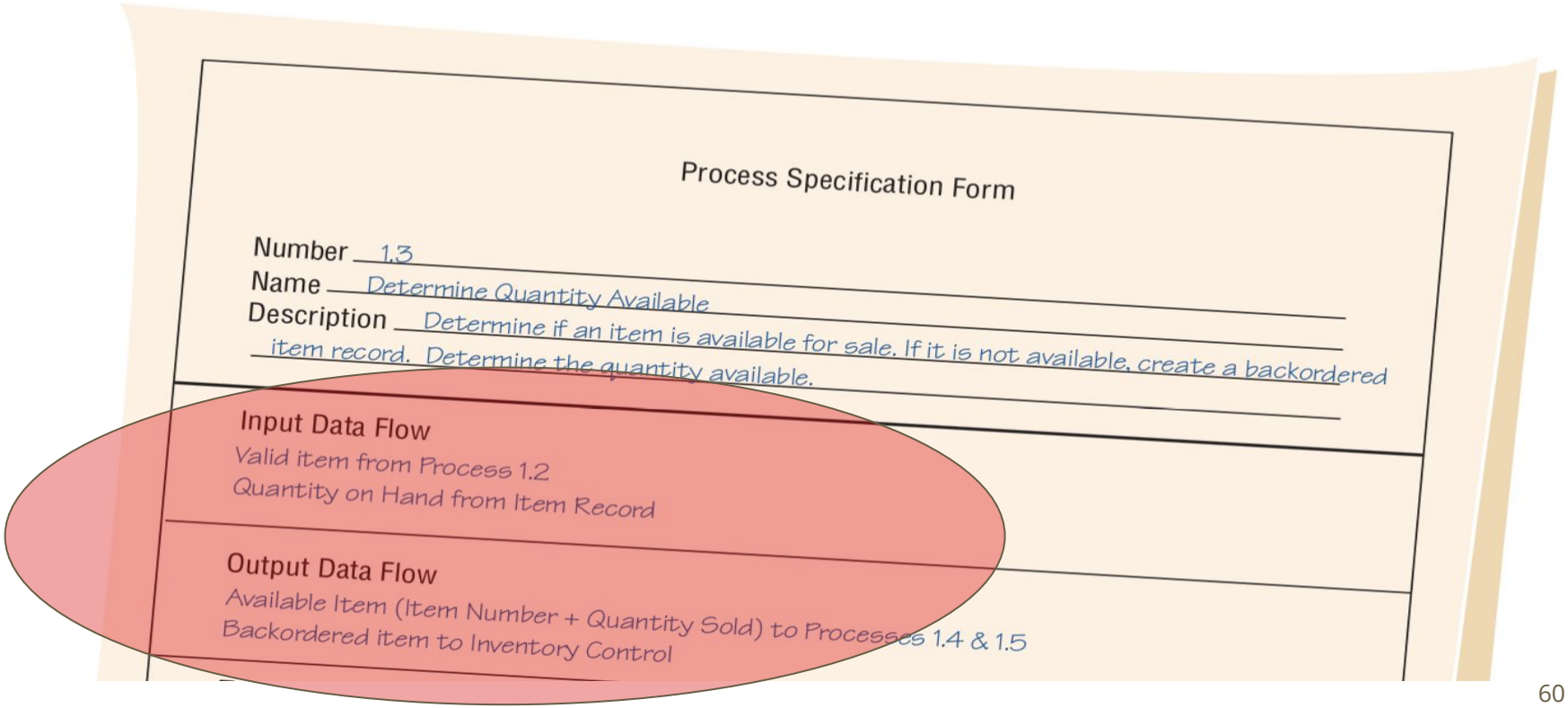
- A method where the process specifications are used to analyze the DFD and data dictionary

- Ensure method logic in process has all the informations required to complete its work

Horizontal Balancing Rules

- All output data flow elements must be either present on input data elements or described in the process logic

Horizontal Balancing



Process Specification Form

Number 1.3

Name Determine Quantity Available

Description Determine if an item is available for sale. If it is not available, create a backordered item record. Determine the quantity available.

Input Data Flow

Valid item from Process 1.2

Quantity on Hand from Item Record

Output Data Flow

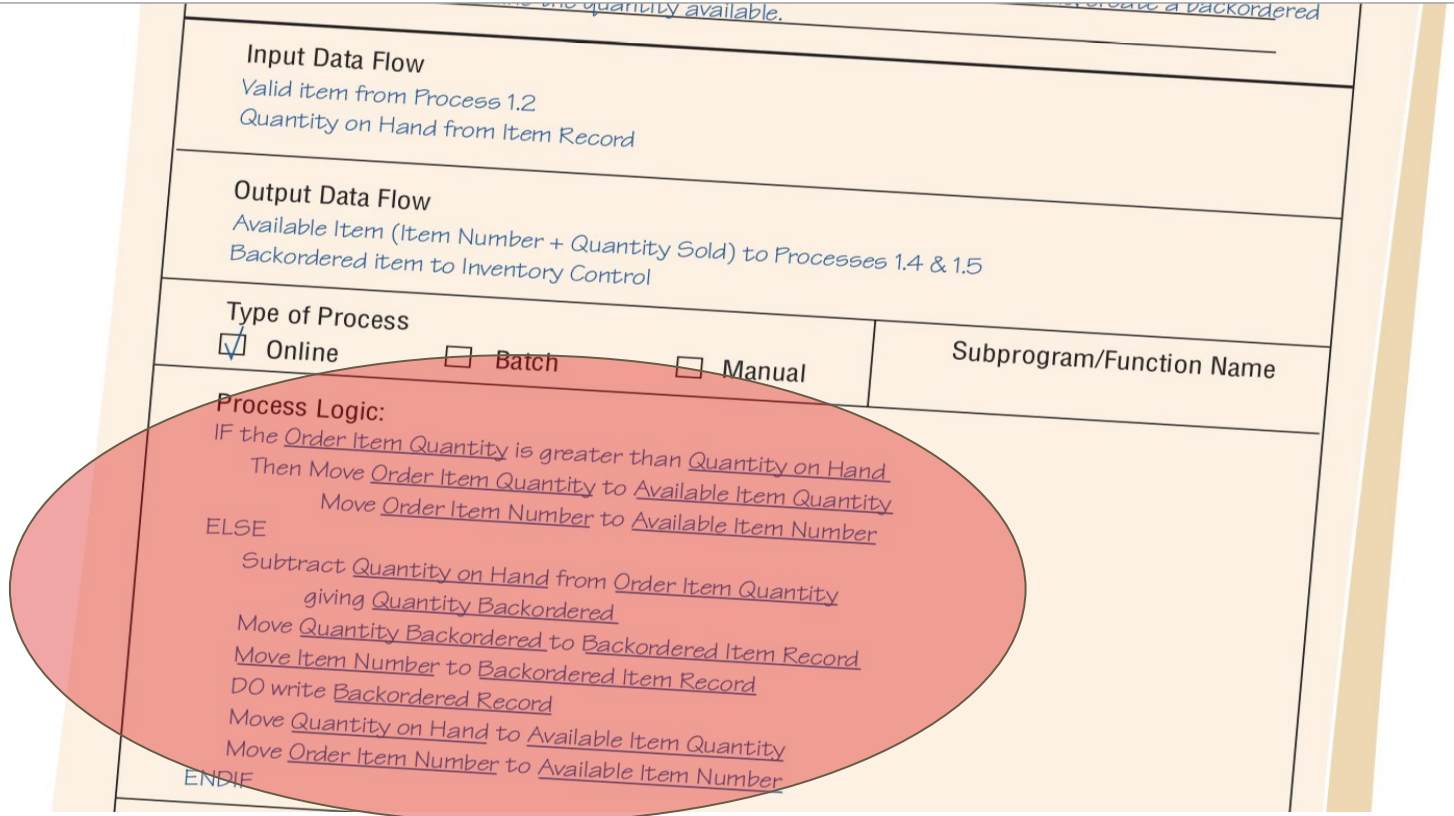
Available Item (Item Number + Quantity Sold) to Processes 1.4 & 1.5

Backordered item to Inventory Control

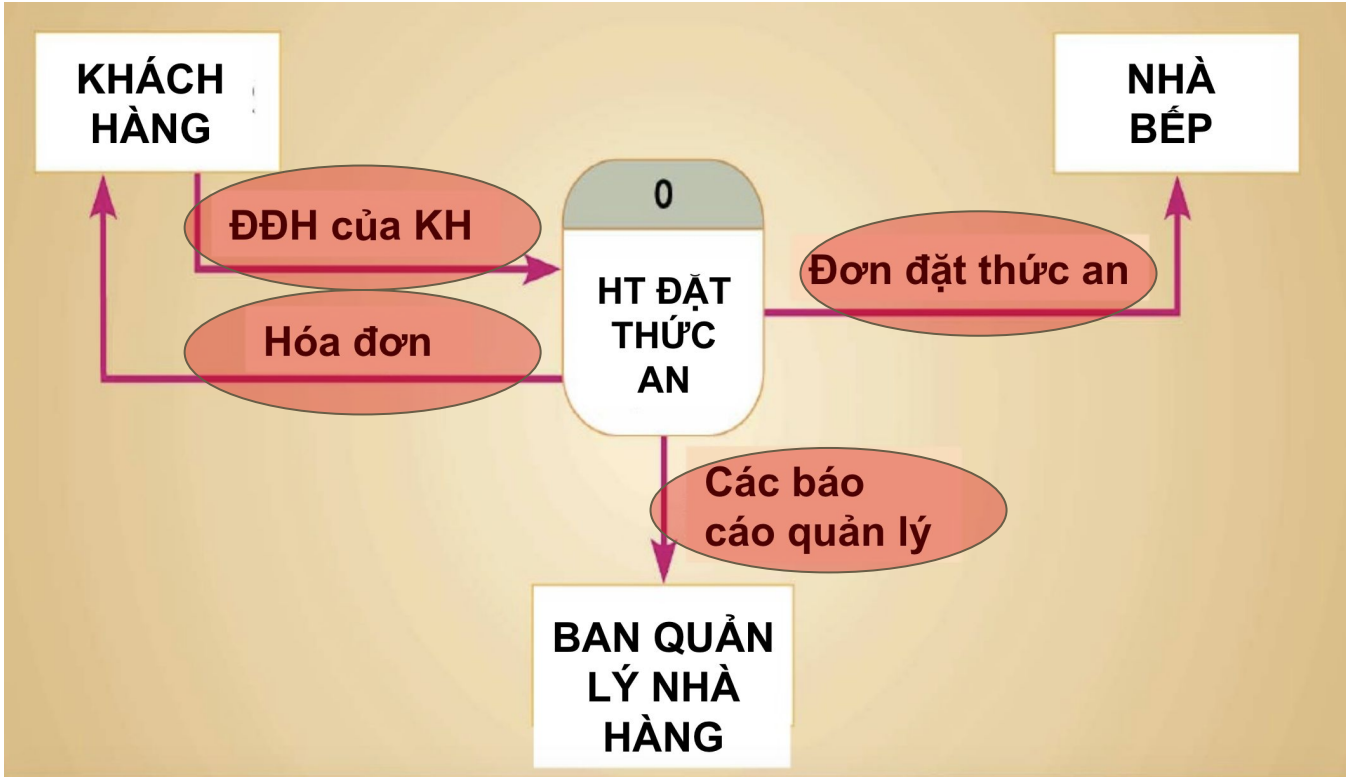
Horizontal Balancing Rules

- Process has the formulas and logic necessary to produce the output

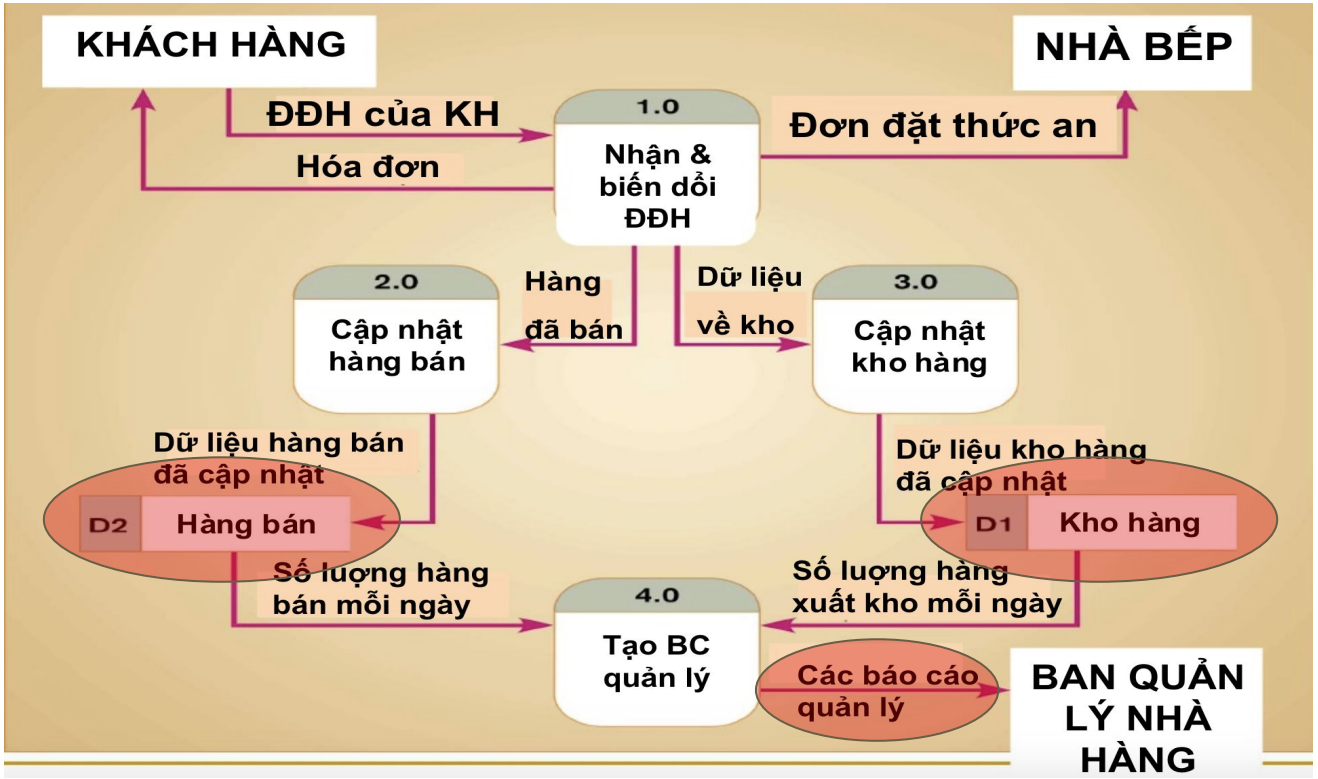
Horizontal Balancing



Horizontal Balancing



Horizontal Balancing



Summary

- Process specifications
  - Structured English
  - Decision tables
  - Decision trees
- Horizontal balancing

Thank you

Any question?