แก้ไขครั้งล่าสุดเมื่อวันที่ 25 กุมภาพันธ์ 2568

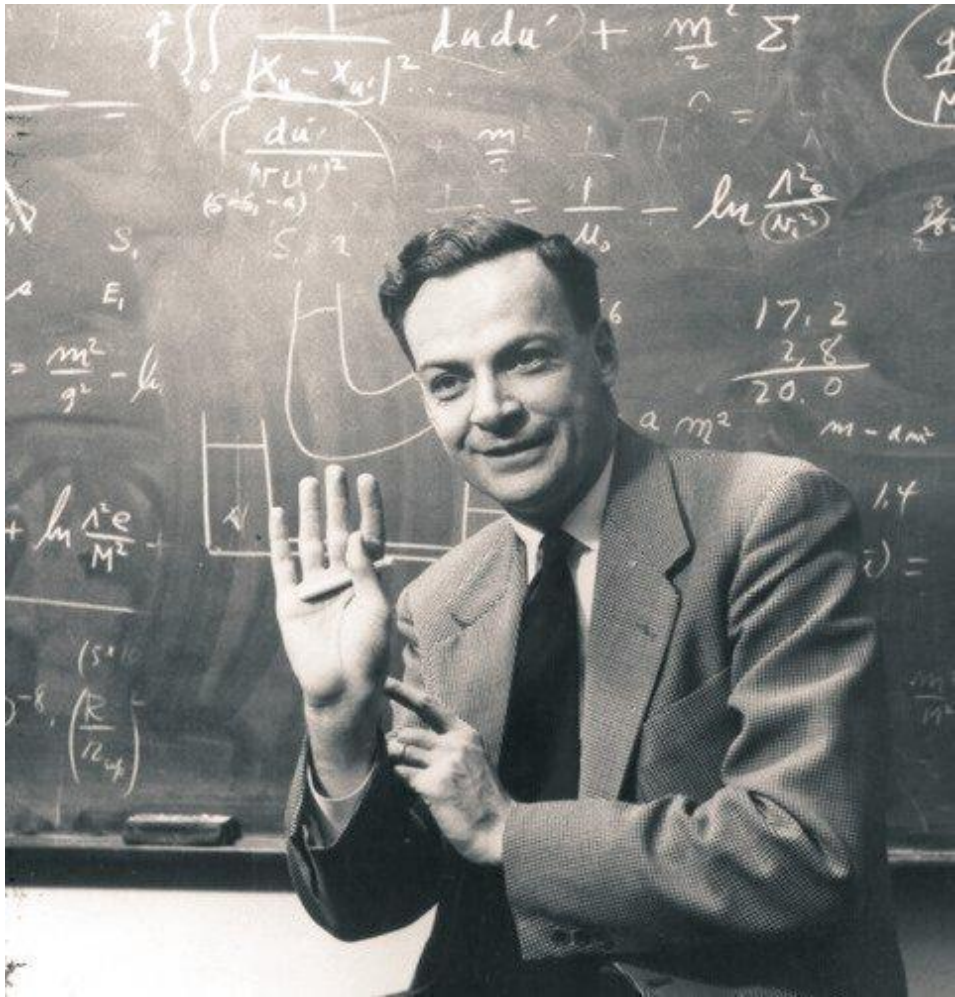
2301361

SYSTEMS ANALYSIS AND DESIGN

4

Understanding Processes with Use Cases and Process Models

If you cannot explain something in simple terms, you don't understand it. (Richard Feynman)



Use Case

Use Case Name: Create preliminary custom drone order	**ID:** UC-6	**Priority:** High
Actor: Customer A person, hardware, software, or user role		
Description: The customer selects and customizes a commercial drone to purchase		
Trigger: Customer wants to purchase a commercial drone		
Type: ☑ External ☐ Temporal		
Preconditions: 1. The customer is authenticated by logging in to his account 2. The Sales System Order Processing application is online		
Normal Course: Happy path 1.0 Order a customized drone 1. The customer selects a base model drone from a list of models 2. The system provides availability status for that model (in stock, out of stock) 3. For out of stock status, system displays expected date available a. Customer accepts future availability date; proceed to step 4 b. Customer rejects future availability date; return to step 1 4. The system displays a list of options and upgrades for the selected model 5. The customer selects desired model options and upgrades 6. Preliminary order with cost estimate is created and displayed 7. Customer may return to step 4, confirm order, save for future consideration, or exit without saving 8. Unconfirmed orders are stored in Unconfirmed Custom Order datastore 9. Confirmed orders are saved in Confirmed Custom Order datastore 10. Shop manager is notified of Confirmed Order requiring approval		
Postconditions: 1. Unconfirmed order is stored in Unconfirmed Custom Order datastore 2. Confirmed order is stored in Confirmed Custom Order datastore 3. Shop manager sent notice of Confirmed Order requiring approval		

ถ้าพัฒนาแบบ Agile ไม่ต้องทำเอกสารนี้ละเอียดมาก
เพราะ user representatives อยู่กับทีมพัฒนาตลอดเวลา
แต่ในบางครั้ง user representatives ก็ไม่ได้อยู่ด้วย

FIGURE 4-1 Create preliminary custom drone order use case—casual format.
A template for this figure is available on the student website.

Figure 4-1 is unwieldy.

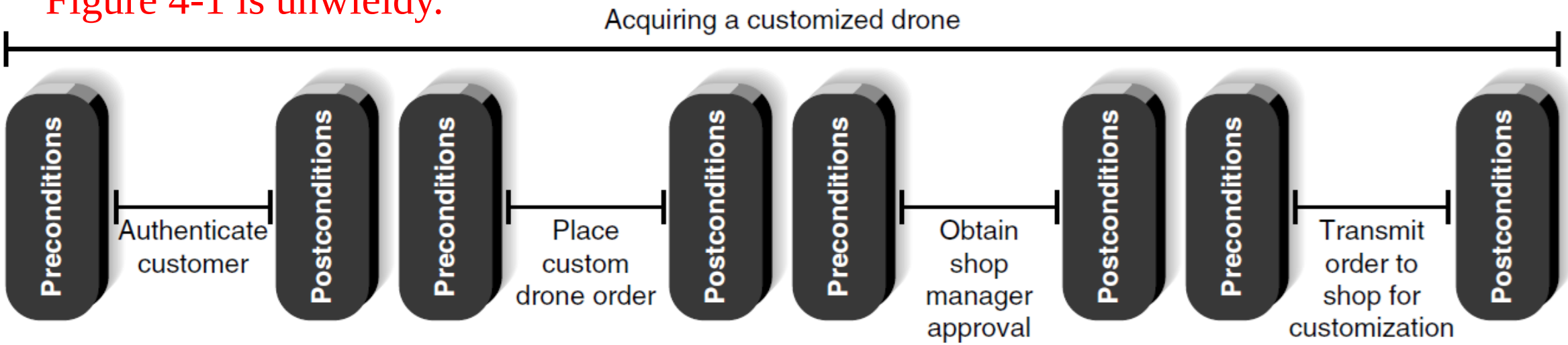


FIGURE 4-2 Chain of use cases with boundaries.

Additional Use Case Issues

Some organizations may choose to include additional sections on their use case forms. If appropriate, it may be helpful to include sections devoted to:

- Alternative paths
- Summary of inputs and outputs
- Frequency of use
- Business rules
- Special requirements
- Assumptions
- Notes and issues

ข้อคิดในการทำ Use Cases

- Use cases depict user–system interactions as abstract, technology-independent steps.
- You should not expect to create a perfect use case on the first try. The process of building use cases is one of gradual refinement: As users and analysts work through the parts of the use case, they often return to previous parts to correct them.
- On the team, there should be at least one person who has a programming perspective to ensure adequate precision and accuracy in the use case; another person who has deep knowledge of the business rules that the system must enforce; and another person who is thoroughly familiar with how the system will actually be used.
- It is tempting for novice analysts, however, to incorrectly assume that the use case is all that is needed to fully define what the system must do.
- By studying the use cases and the functional requirements derived from them, the testing personnel can readily identify elements of the tests they will want to perform when the system enters testing.

- The system displays a list of base drone models
- The system accepts customer selection of base drone model
- The system displays in stock/out of stock status for selected drone model
- For an out-of-stock model,
  - The system displays the expected date of availability
  - The system asks customer to accept future date available and continue or to select a different drone model
- The system displays optional features for the selected drone model (batteries, motors, cameras, sensors, etc.)
- The system accepts user choices of options
- The system displays summary and price of selected drone configuration
- The system allows user to continue modifying the drone configuration, save the order for later, confirm the order, or exit without saving.
- For orders saved without confirming, the system stores the order in the Unconfirmed Custom Order datastore
- For confirmed orders,
  - The system displays a completed order summary for the customer
  - The system stores the order in the Confirmed Custom order
  - The system sends a notice of new Confirmed Custom order to the Shop Manager for approval

FIGURE 4-3 Place custom drone order (normal course) functional requirements.

Step	Activities	Typical Questions Asked[a]
1. Identify the use cases.	Start a use case report form for each use case by filling in the name, description, and trigger. If there are more than nine use cases, group them into packages.	Ask *who*, *what*, *when*, and *where* about the use cases (or tasks). What are the major tasks that are performed? What triggers this task? What tells you to perform this task?
2. Identify the major steps within each use case.	For each use case, fill in the major steps needed to complete the task.	Ask *how* about each use case. What information/forms/reports do you need to perform this task? Who gives you these information/forms/reports? What information/forms/report does this produce and where do they go? How do you produce this report? How do you change the information on the report? How do you process forms? What tools do you use to do this step (e.g., paper, e-mail, phone)?

FIGURE 4-4
Steps for writing for use cases.

3.	Identify elements within steps.	For each step, identify its triggers and its inputs and outputs.	Ask *how* about each step. How does the person know when to perform this step? What forms/reports/data does this step produce? What forms/reports/data does this step need? What happens when this form/report/data is not available?
4.	Confirm the use case.	For each use case, validate that it is correct and complete.	Ask the user to execute the process, using the written steps in the use case—that is, have the user role-play บทบาทสมมติ the use case.

FIGURE 4-4
Steps for writing for use cases.

In this example, we focus on the Drone Customization Shop Management.

Event	Response
1) New shop work order is received	• Shop work order arrival time is recorded
2) Shop work order is assigned to qualified technician	• Work order added to technician's work assignment
3) Parts Request for base model drone and additional components generated	• Parts request stored • Inventory department receives Parts Request
4) Requested parts arrive in Shop Parts room	• Parts arrival recorded on Parts Request
5) All parts on a Parts Request are available	• Technician notified that all parts are available
6) Technician records date/time work begins on work order	• Shop work order updated with start date/time
7) Technician records completion of work on work order	• Shop work order updated with comple-tion date/time • Customer is notified of order completion

FIGURE 4-5 Sample event-response list. A template for this figure is available on the student website.

these events are three parts of the overall user goal of obtaining all needed components for a drone customization work order.

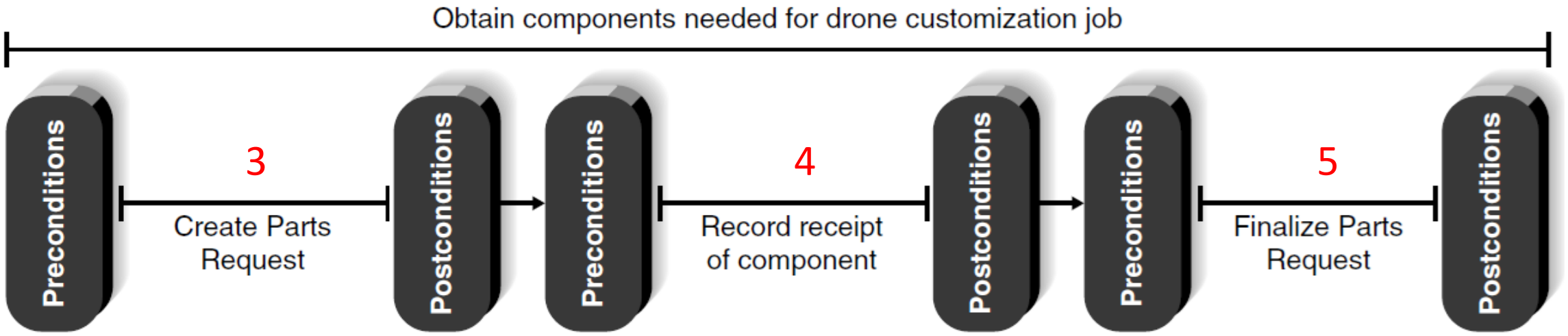


FIGURE 4-6 Chain of use cases for obtaining drone components.

If the project team discovers more than eight or nine major use cases, this suggests that the system is complex (or that the use cases are not defined at the right level of detail). If there really are more than eight or nine major use cases, the use cases are grouped together into **use case packages**.

ใน data flow diagram น่าจะยุบเป็น process ใหญ่ชื่อ Obtain all needed components (for a drone customization work order) แล้วแตกเป็น process ย่อย 3 processes

1. Create parts request
2. Record receipt of component
3. Finalize parts request

Major use cases with basic information

Use Case Name: Create Parts Request	**ID:** UC-3	**Priority:** High
Actor: Shop Manager		
Description: This use case describes how the Shop Manager creates a Parts Request.		
Trigger: Shop manager receives notice of new shop work order arrival from Sales System. **Type:** ☑ **External** ☐ **Temporal**		
Preconditions: Shop manager is authenticated Parts Request application is available and online Inventory application is available and online		

Use Case Name: Record Receipt of Component	**ID:** UC-4	**Priority:** High
Actor: Shop Parts Room Clerk		
Description: This use case describes how the Parts Room Clerk Records delivery of drone component from Inventory Department.		
Trigger: Drone component arrives in Parts Room from Inventory Department. **Type:** ☑ **External** ☐ **Temporal**		
Preconditions: Parts Room Clerk is authenticated Parts Request application is available and online		

FIGURE 4-7 Major use cases with basic information.

Use Case Name: Finalize Parts Request	**ID:** UC-5	**Priority:** High
Actor: Parts Room Clerk		
Description: This use case describes how the Parts Room Clerk finalizes a Parts Request.		
Trigger: Notification received that all parts are available for a Parts Request. **Type:** ☑ **External** ☐ **Temporal**		
Preconditions: Parts Room Clerk is authenticated All components listed on Parts Request have arrived in Parts Room		

FIGURE 4-7 Major use cases with basic information.

If you end up with more than nine steps or steps that vary greatly in size, you must go back and adjust the steps. Recognizing the size of the steps takes practice but will become natural in time.

After several revisions, the team settled on the partial use cases shown in Figure 4-8.

Use Case Name: Create Parts Request	**ID:** UC-3	**Priority:** High
Actor: Shop Manager		
Description: This use case describes how the Shop Manager creates a Parts Request		
Trigger: Shop Manager receives notice of new shop work order arrival from Sales System		
Type: ☑ External ☐ Temporal		
Preconditions: 1. Shop manager is authenticated 2. Parts Request application is available and online 3. Inventory application is available and online		
Normal Course: 1.0 Create Parts Request 1. Shop manager opens the shop work order 2. Shop manager opens a blank Parts Request 3. For each required component part a. Shop manager enters component part and quantity needed b. System queries the inventory datastore and records part status: (In stock/ Out of stock with expected date available) 4. Shop manager verifies Parts Request is complete 5. System stores new Parts Request 6. System transmits notice of Parts Request to be filled to Inventory Department		**Information for Steps**
Postconditions: 1. New Parts Request record created and stored 2. Inventory Department notified of Parts Request to be filled		

FIGURE 4-8 Major use cases with steps completed.

Use Case Name: Record Receipt of Component	**ID:** UC-4	**Priority:** High
Actor: Shop Parts Room Clerk		
Description: This use case describes how the Shop Parts Room clerk records delivery of component from Inventory department.		
Trigger: Drone component arrives in Parts Room from Inventory department		
Type: ☑ External ☐ Temporal		
Preconditions: 1. Parts room clerk is authenticated 2. Parts Request application is available and online		
Normal Course: 1.0 Record Receipt of Component 1. Parts Room clerk retrieves correct Parts Request 2. Parts Room clerk verifies part is correct and undamaged 3. System records date/time part is received 4. Parts Room clerk records the Parts Room location where the part will be stored until pickup 5. System stores updated Parts Request 6. System checks to see if all parts on the Parts Request are available; if they are, the Parts Room clerk is notified to perform Finalize Parts Request use case	**Information for Steps**	
Postconditions: 1. Parts request updated with received part		

FIGURE 4-8 Major use cases with steps completed.

Use Case Name: Finalize Parts Request	**ID:** UC-5	**Priority:** High
Actor: Shop Parts Room clerk		
Description: This use case describes how the Parts Room clerk finalizes a Parts Request		
Trigger: Notification received that all parts are available for a Parts Request		
Type: ☑ External ☐ Temporal		
Preconditions: 1. Parts room clerk is authenticated 2. All components listed on Parts Request have arrived in Parts Room		
Normal Course: 1.0 Finalize Parts Request 1. Parts room clerk opens the Parts Request and the associated Shop Work Order 2. Parts room clerk verifies that all listed components are currently in the Parts Room 3. Parts room clerk changes the Parts Request Status to "complete" 4. System records the completion date/time in the Shop Work Order 5. System notifies the assigned technician that job is ready		**Information for Steps**
Postconditions: 1. Parts Request status is complete 2. Date/time all parts are available recorded in Shop Work Order 3. Technician notified that job is ready		

The use case forms in Figure 4-8 require some final work before they are complete. The last column ("Information for Steps") must be completed, and arrows may be drawn to describe **inputs** and **outputs** from the steps.

FIGURE 4-8 Major use cases with steps completed.

Use Case Name: Create Parts Request	**ID:** UC-3	**Priority:** High

Actor: Shop Manager

Description: This use case describes how the Shop Manager creates a Parts Request

Trigger: Shop Manager receives notice of new shop work order arrival from Sales System

Type: ☑ External ☐ Temporal

Preconditions:
1. Shop manager is authenticated
2. Parts Request application is available and on-line
3. Inventory application is available and on-line

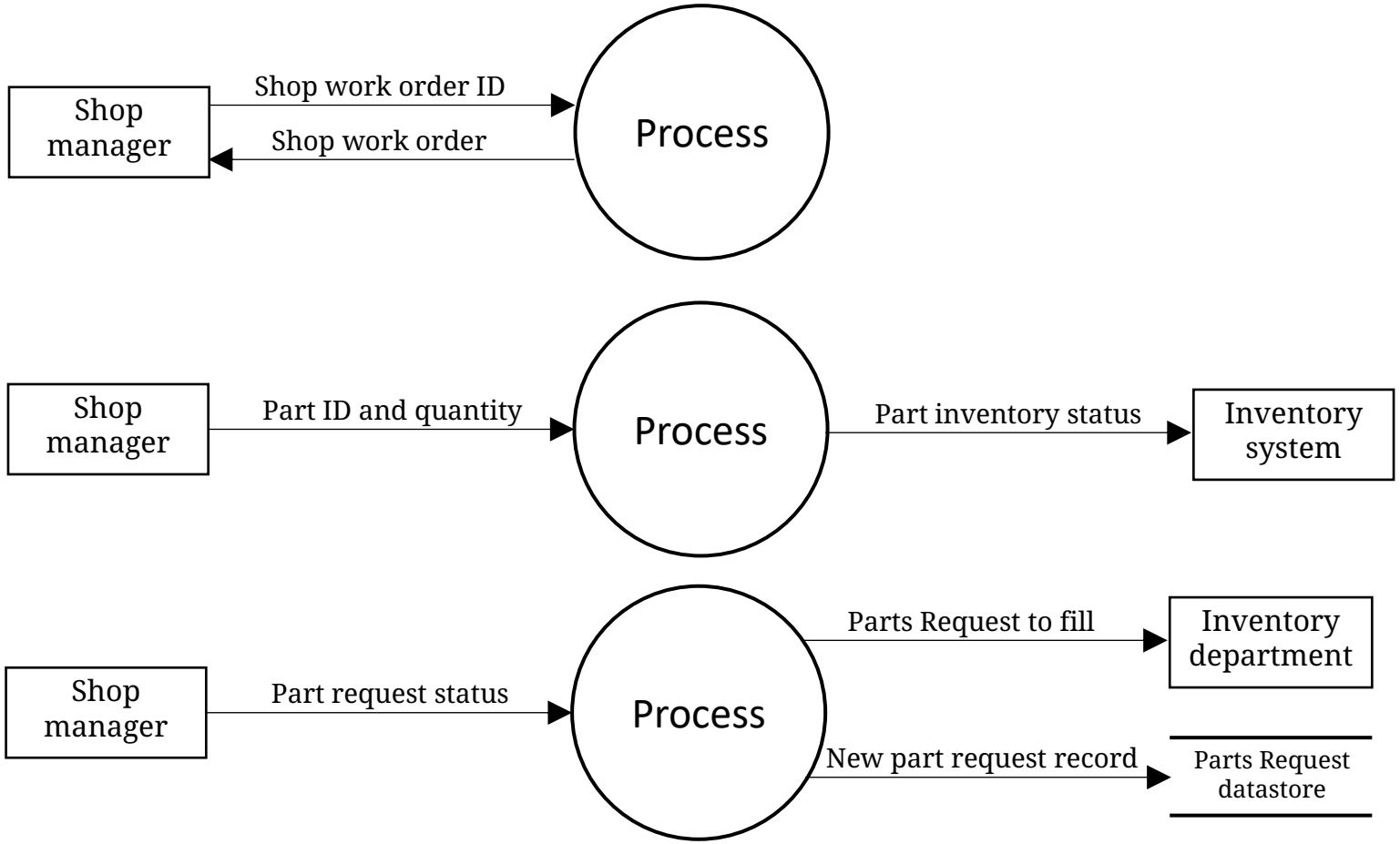
Normal Course:

	Information for Steps
1.0 Create Parts Request	→ Shop work order ID
1. Shop manager retrieves the shop work order	← Shop work order
2. Shop manager opens a blank Parts Request	
3. For each required component part	
a. Shop manager enters component part and quantity needed	→ Part ID, quantity needed
b. System queries the inventory datastore and records part status: (In stock/ Out of stock with expected date available)	← Part status in inventory
4. Shop manager verifies Parts Request is complete	→ Part Request status
5. System stores new Parts Request	→ New Parts Request record
6. System transmits notice of Parts Request to be filled to Inventory Department	→ Parts Request to fill

Postconditions:
1. New Parts Request record created and stored
2. Inventory Department notified of Parts Request to be filled

Summary Inputs	Source	Summary Outputs	Destination
Shop work order ID	Shop manager	Shop work order	Shop manager
Part ID and quantity	Shop manager	Part inventory status	Inventory system
Part request status	Shop manager	New part request record	Parts Request datastore
		Parts Request to fill	Inventory department

Summary Inputs	Source	Summary Outputs	Destination
Shop work order ID	Shop manager	Shop work order	Shop manager
Part ID and quantity	Shop manager	Part inventory status	Inventory system
Part request status	Shop manager	New part request record	Parts Request datastore
		Parts Request to fill	Inventory department



The Summary area for inputs and outputs found at the end of the use case form is completed once the team is satisfied with the steps, inflows, and outflows listed previously. In this section, all the input flows are listed in the left-most column and their source is specified in the adjacent column. In the third column, all the output flows are listed, and their destination is specified in the right-most column. As we have mentioned, this summary area allows the team to easily view all the inputs that must be included to complete the use case and all the outputs that will be produced by the use case. This area of the use case form will be especially useful if the team decides to depict the system with **data flow diagrams**.

เลือกใช้ diagram อื่น ๆ ที่เหมาะสมกับระบบ Data Flow Diagram (DFD) เหมาะกับงานภายในองค์กร (ธุรกิจเอกชน/ราชการ) แต่ไม่เหมาะกับวีดีโอเกม เกมออนไลน์ ฯลฯ

YOUR TURN 4-1 Campus Housing

Create a set of use cases for the following high-level requirements in a housing system run by the Campus Housing Service. The Campus Housing Service helps students find apartments. Owners of apartments fill in information forms about the rental units they have available (e.g., location, number of bedrooms, monthly rent), which are entered into a database. Students can search through this database via the Web to find apartments that meet their needs (e.g., a two-bedroom apartment for $800 or less per month within ½ mile of campus). They then contact the apartment owners directly to see the apartment and possibly rent it. Apartment owners call the service to delete their listing when they have rented their apartment(s).

In building the major use cases, follow the four-step process: identify the use cases, identify the steps within them, identify the elements within the steps, and confirm the use cases.

CONCEPTS IN ACTION 4-A | Building a Bad System?

Several years ago, a well-known national real estate company built a computer-based system to help its real estate agents sell houses more quickly. The system, which worked in many ways like an early version of realtor.com, enabled its agents to search the database of houses for sale to find houses matching the buyer's criteria using a much easier interface than the traditional system. The system also enabled the agent to show the buyer a virtual tour of selected houses listed by the company itself. It was believed that by more quickly finding a small set of houses more closely matching the buyer's desires, and by providing a virtual tour, the buyers (and the agent) would waste less time looking at unappealing houses. This would result in happier buyers and in agents who were able to close sales more quickly, leading to more sales for the company and higher commissions for the agent.

The system was designed with input from agents from around the country and was launched with great hoopla. The initial training of agents met with a surge of interest and satisfaction among the agents, and the project team received many congratulations.

Six months later, satisfaction with the system had dropped dramatically, absenteeism had increased by 300%, and agents were quitting in record numbers; turnover among agents had risen by 500%, and in exit interviews, many agents mentioned the system as the primary reason for leaving. The company responded by eliminating the system—with great embarrassment.

One of an agent's key skills was the ability to find houses that match the buyer's needs. The system destroyed the value of this skill by providing a system that could enable less skilled agents to perform almost as well as highly skilled ones. Worse still—from the viewpoint of the agent—the buyer could interact directly with the system, thus bypassing the "expertise" of the agent.

Questions

1. How were the problems with the system missed?
2. How might these problems have been foreseen and possibly avoided?
3. In perfect hindsight, given the widespread availability of such systems on the Internet today, what should the company have done?

Adapted from: "The Hidden Minefields in Sales Force Automation Technologies," *Journal of Marketing*, July 2002, by C. Speier and V. Venkatesh.

ทำ Use Cases ที่เหลือให้ครบ

Event	Response	Requirements
• Client wants to learn about services	• Flight services and data analyses displayed	1.1–1.3
• Pilots wants to learn about Pilot Partnership program	• Pilot Partnership details displayed	5.1–5.2
• Client creates account	• New client account and profile created	2.1–2.3
• Pilot creates Pilot Partnership Agreement	• Pilot Partnership completed	6.1–6.4
• Client requests drone flight	• Open Drone Flight Request created	3.1–3.2
• Pilot receives drone flight assignment	• Flight assigned to pilot	7.1–7.4, 3.3
• Client wants to view status of open flight requests	• Flight request status displayed	4.1
• Flight is completed	• Flight sensor data uploaded; data analyses initiated; customer notified; flight request closed	8.1–8.3, 4.2
• Client requests more data analysis for completed flights	• Data analyses initiated	4.5

FIGURE 4-10
DrōnTeq Client Services event-response list.

Use Case Name: Notify Pilots of New Flight Request	**ID:** UC-2A	**Priority:** High

Actor: Flight Request system

Description: This use case describes how the system notifies pilots of new flight request

Trigger: New Flight Request is submitted by client (described in UC-1 (Create Flight Request)

Type: ☑ External ☐ Temporal

Preconditions:
1. New Flight Request is submitted and confirmed by client
2. Flight Assignment application is available and online

Normal Course:

	Information for Steps
1.0 Notify pilots of new flight request	
1. System obtains the flight request	← Flight request
2. System uses the latitude/longitude of the flight area to define the flight proximity region	← Flight location details
3. System develops a list of pilots in the flight proximity region	← Pilot locations
4. System develops price guidelines for the flight based on the characteristics of the flight request	← Flight details ← Pricing guidelines
5. System prepares a flight request notification for all pilots in the flight proximity region, including location, requested flight features, price guidelines, and opening/closing date/time of the pilot bidding window.	
6. System transmits flight request notification to all pilots in the flight proximity region	→ Flight Request Notification message
7. System stores new flight request notification	→ New Flight Request Notification Record
8. Pilot dashboards for all pilots receiving the flight notification are modified to include the flight request notification with a bidding window countdown clock.	→ New pilot dashboard notice

Postconditions:
1. New flight request notification record created and stored
2. Pilots in the flight proximity region receive flight notification message
3. Pilot dashboards for the notified pilots are modified

Exceptions:
E1. No pilots are found in the flight proximity region (occurs at step 3)
1. System increases the radius of the distance from the flight location by 25 miles
2. Return to step 2, Normal Course, to recalculate the flight proximity region

Summary Inputs	Source	Summary Outputs	Destination
Flight Request	Create Flight Request use case	Flight Request Notification message	Pilots
Flight location	Flight request	New Flight Request	Flight Request
Pilot locations	Pilot datastore	Notification record	Notification datastore
Flight details	Flight request	Pilot dashboard update	Dashboard of notified pilots
Pricing guidelines	Price guideline datastore		

FIGURE 4-11 DrōnTeq use cases.

Use Case Name: Pilot Submits Bid	**ID:** UC-2B	**Priority:** High

Actor: Pilot

Description: This use case describes how a pilot submits a bid for an open flight request

Trigger: Pilot received notification of new flight request

Type: ☑ External ☐ Temporal

Preconditions:
1. Pilot is authenticated and signed in to his/her dashboard
2. Open flight request is displayed on pilot dashboard
3. Flight Request application is available and online

Normal Course:	**Information for Steps**
1.0 Submit Pilot Bid	
1. Pilot selects the new flight request displayed on dashboard	← Flight Request
2. System displays details of the flight request including location, requested flight features, price guidelines, and opening/closing date/time of the pilot bidding window.	← Flight request notification details
3. Pilot selects "Submit a bid" option	
4. System displays Flight bid form	
5. Pilot enters bid price and planned date/time for flight	← Flight Bid details
6. System verifies that the bid meets the terms of the flight request If no errors, continue; if errors, display error message and return to step 5	← Flight Request details
7. System requests pilot confirmation of the bid	← Pilot confirmation
8. System stores new Flight Bid record	→ New flight bid record

Postconditions:
1. New flight bid is stored

Exceptions:
E1: Bidding window is closed (occurs at step 7)
1. Date/time of pilot confirmation is after the closing date/time of the bidding window
2. Modify status of flight bid to "invalid/late"
3. Save flight bid record
4. Notify pilot of late submission bid status

Summary Inputs	Source	Summary Outputs	Destination
Flight Request record	Flight request datastore	Flight bid record	Flight bid datastore
Flight request notification details	Flight request notification datastore	Flight bid status	Pilot
Flight bid details	Pilot		
Bid confirmation	Pilot		

FIGURE 4-11 *Continued*

Use Case Name: Select Winning Flight Bid	**ID:** UC-2C	**Priority:** High

Actor: Flight Operations Manager

Description: This use case describes how the Flight Operations Manager finalizes the selection of the winning pilot bid on a flight

Trigger: A flight bid window has closed

Type: ☐ External ☑ Temporal

Preconditions:
1. Flight Operations Manager is authenticated
2. Bidding window on a flight request has closed

Normal Course:	**Information for Steps**
1.0 Select Winning Flight Bid	
1. System posts "closed to bid" message on pilot's dashboard	→ Pilot dashboard update
2. Flight Operations manager requests list of all bids for the flight	→ Flight Bid List Request
3. System displays list of all bids	← Flight Bid records
4. For each bid,	← Pilot's drone details
a. The flight manager verifies the drone's capability to perform the flight	← Flight Request details
b. Bids based on drone without required equipment are marked "Not Qualified," updated, and removed from bid list	→ Updated Flight Bid record
5. System sorts and ranks remaining bids based on flight completion time and bid price	← Qualified Flight Bid records
6. Flight Operations Manager selects bid that optimizes flight completion time and bid price	→ Updated Flight Bid record
7. System changes status of all unselected flight bids	→ Updated Flight Bid records
8. System sends notification message to selected bid pilot	→ Bid Award Notification
9. System posts flight details on selected pilot's dashboard	→ Awarded flight details
10. System notifies all other bidding pilots of the final selection	→ Unsuccessful bid notice
11. System notifies customer that flight has been assigned to a pilot	→ Flight Assigned notice
12. System updates Flight Request with flight assignment details.	→ Updated Flight Request

Postconditions:
1. Pilot of selected bid is notified
2. Selected pilot's dashboard is updated with flight details
3. Pilots' bids not selected are notified
4. Flight bid records selection status is updated
5. Client notified of flight assignment
6. Flight Request record updated with flight assignment

Exceptions:
E1: No bids submitted (occurs at Step 2)
1. Flight operations manager modifies flight proximity area and/or price guidelines
2. Flight operations manager performs Use Case 2A to re-post modified Flight Request
3. If Flight Request has been posted twice with no bids, the Flight Operations manager will contact the client; exit the use case

Summary Inputs	Source	Summary Outputs	Destination
Flight Bid records	Flight Bid datastore	Bidding window closed	Pilot dashboards
Pilot drone details	Pilot drone datastore	Updated Flight Bids	Flight Bid datastore
Flight Request details	Flight Request datastore	Bid Award Notification	Selected Pilot
Qualified Flight Bids	Flight Bid datastore	Awarded Flight details	Selected Pilot dashboard
		Unsuccessful bid notice	Unselected pilots
		Assigned flight notice	Flight Client
		Assigned Flight Request	Flight Request datastore

FIGURE 4-11 *Continued*

Data Flow Diagrams (DFDs)

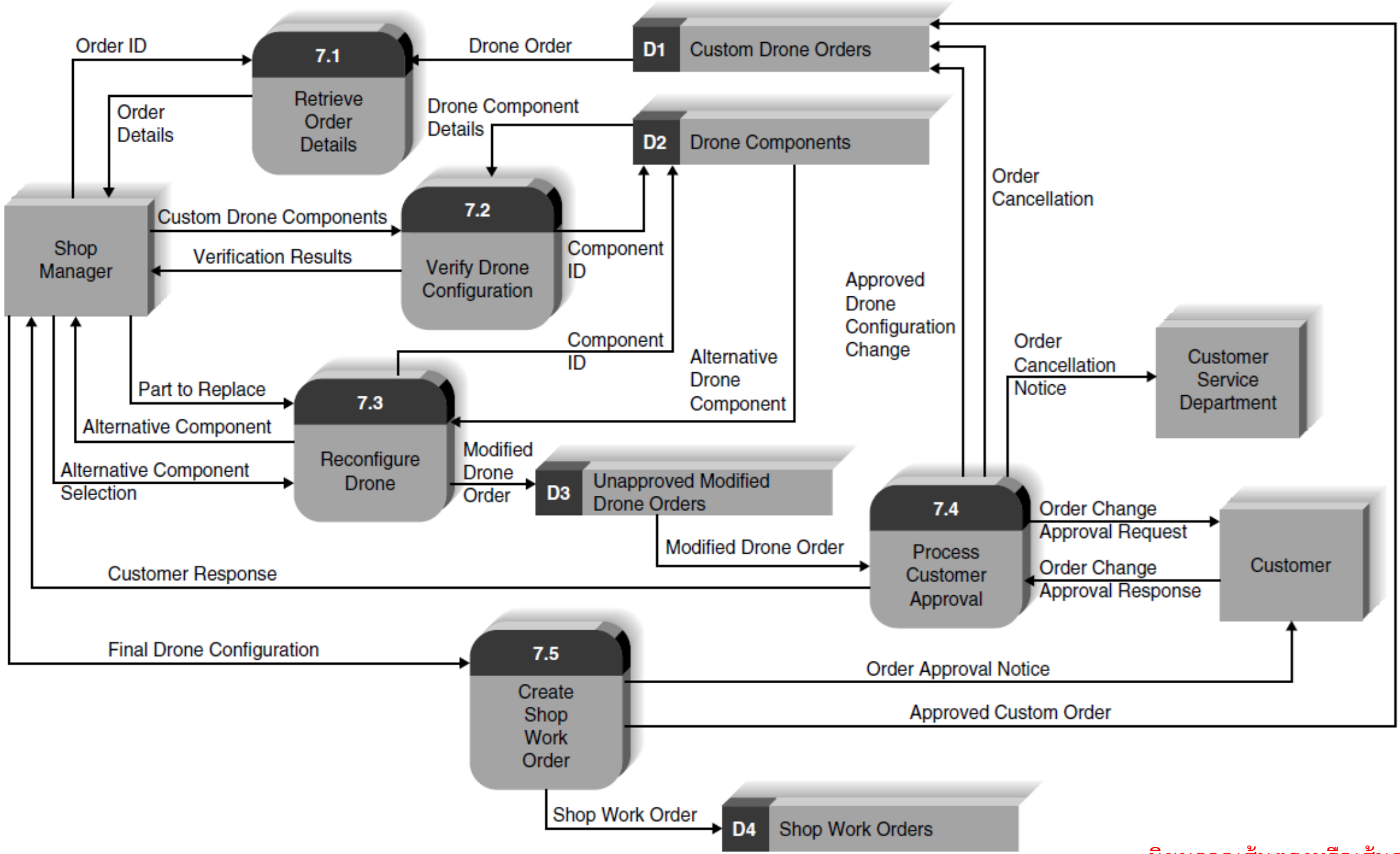


FIGURE 4-12 Shop manager approval of custom drone order level 1 DFD.

นิยมลากเส้นตรงหรือเส้นฉากเท่านั้น
ไม่ลากเส้นโค้ง

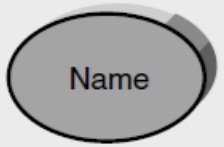
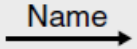
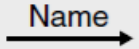
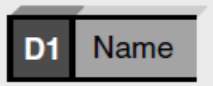
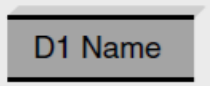
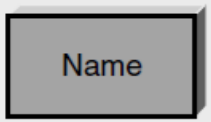
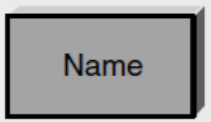
Data Flow Diagram Element	Typical Computer-Aided Software Engineering Fields	Gane and Sarson Symbol	DeMarco and Yourdon Symbol
Every *process* has a number a name (verb phase) a description at least one output data flow at least one input data flow	Label (name) Type (process) Description (what is it) Process number Process description (structured English) Notes		
Every *data flow* has a name (a noun) a description one or more connections to a process	Label (name) Type (flow) Description Alias (another name) Composition (description of data elements) Notes		
Every *data store* has a number a name (a noun) a description one or more input data flows one or more output data flows	Label (name) Type (store) Description Alias (another name) Composition (description of data elements) Notes		
Every *external entity* has a name (a noun) a description	Label (name) Type (entity) Description Alias (another name) Entity description Notes		

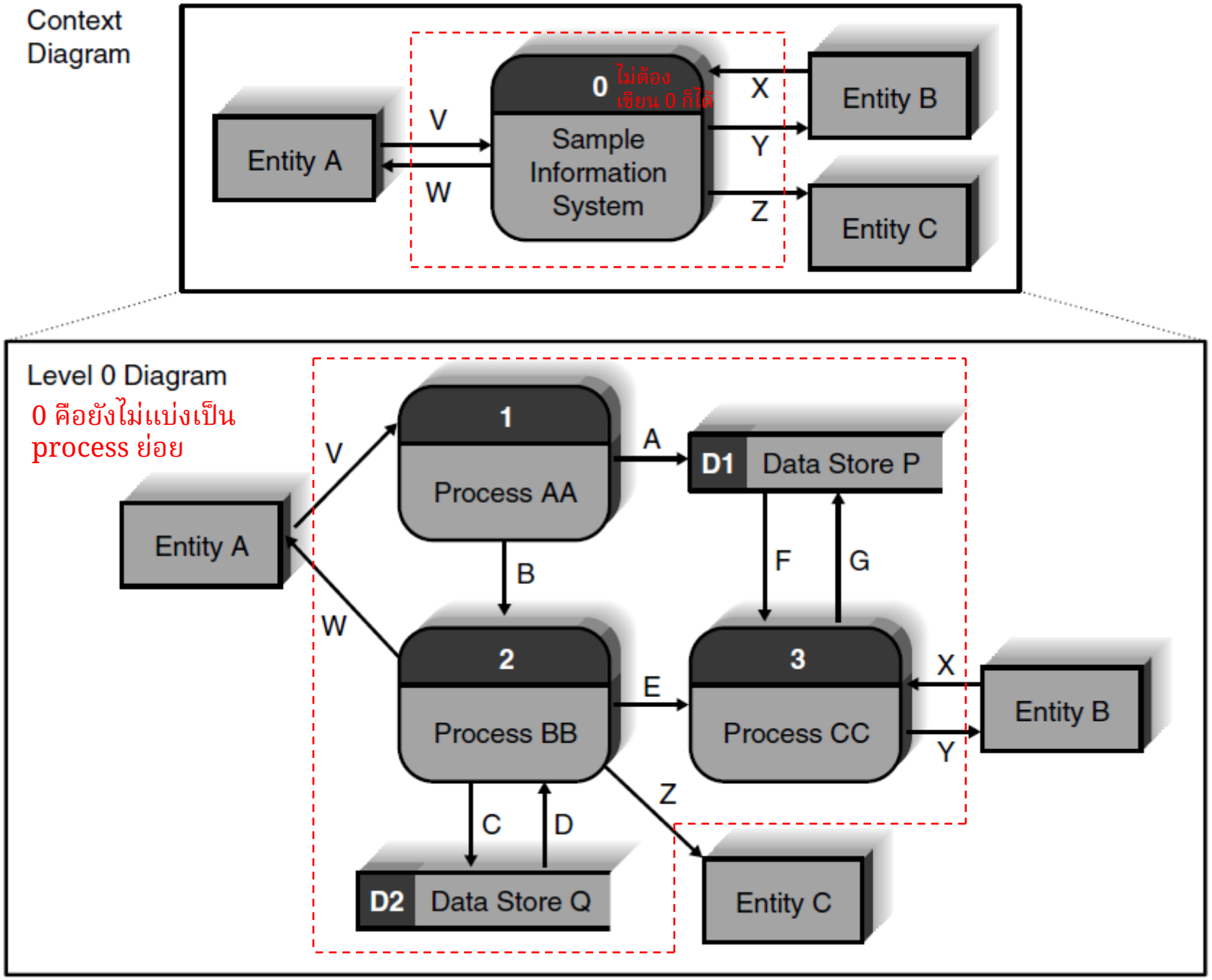
FIGURE 4-13 Data flow diagram elements.

A **process** is an activity or a function that is performed for some specific business reason. Processes can be manual or computerized.

A **data flow** is a single fact, such as Order ID (sometimes called a data element), or a logical collection of several facts (e.g., new shop work order). Every data flow should be named with a noun.

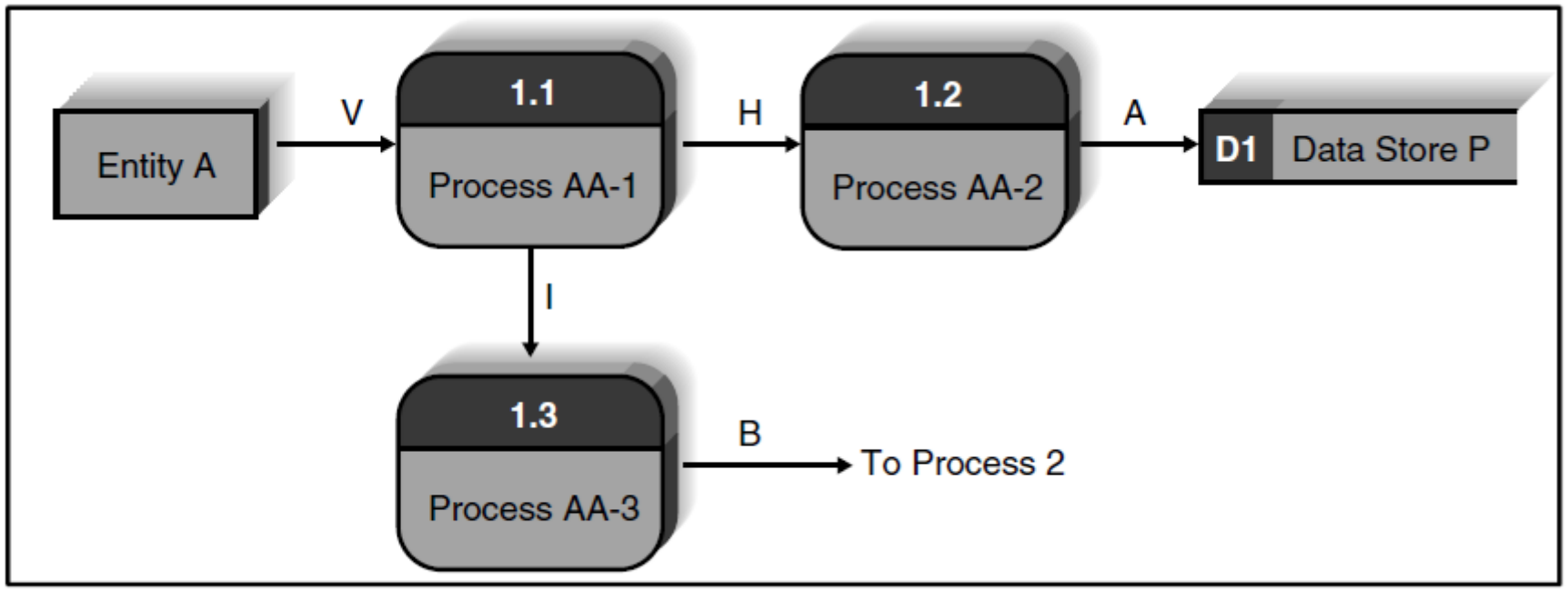
A **data store** is a collection of data that is stored in some way (which is determined later when creating the physical model). Every data store is named with a noun and is assigned an identification number and a description.

An **external entity** is a person, organization, organization unit, or system that is external to the system, but interacts with it (e.g., customer, clearinghouse, government organization, accounting system).



ปกติผมทำ level 0 ก่อน แล้วค่อยทำ level 1, 2, 3, ... context diagram ไว้ทำทีหลังสุด แยกเขียนเป็นหลาย ๆ หน้าได้

FIGURE 4-14
Context diagram decomposed into level 0 diagram.



Process 1 Level 1 Diagram

Level 1 คือแบ่ง 1 ครั้ง
ทศนิยม 1 ตำแหน่ง

นิสิตมักจะทำผิด คือพยายาม flow data จาก process แรกไปจนถึง process สุดท้าย หรือให้มีเส้นเชื่อมทุก process เข้าด้วยกัน
วิธีที่ถูกคือ process แรกอาจจะรับข้อมูลจากผู้ใช้ แล้วบันทึกลง data store จากนั้น process ต่อ ๆ ไป ก็อ่านจาก data store

FIGURE 4-15
Processes 1, 2, and 3 level 1 diagrams.

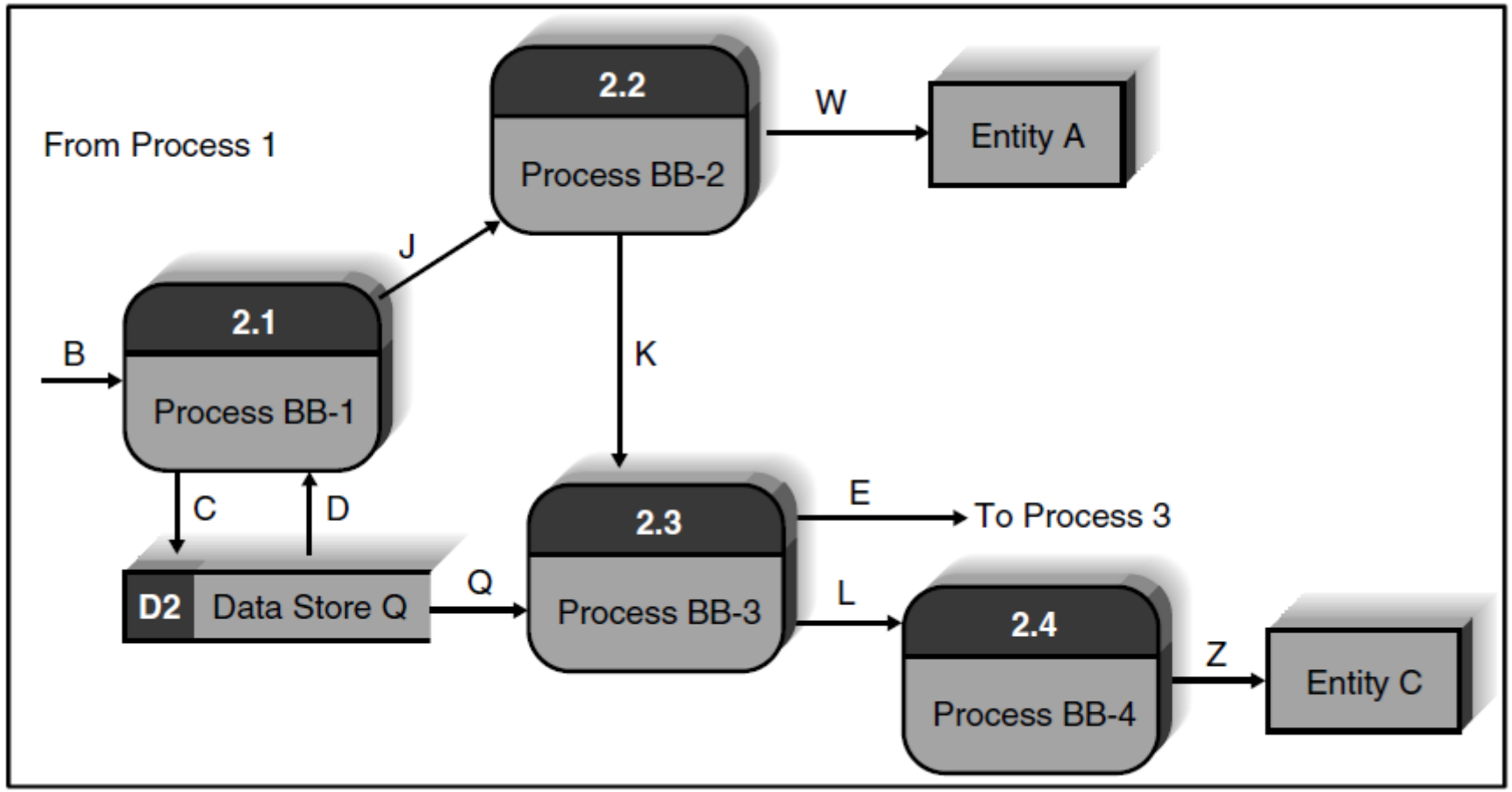
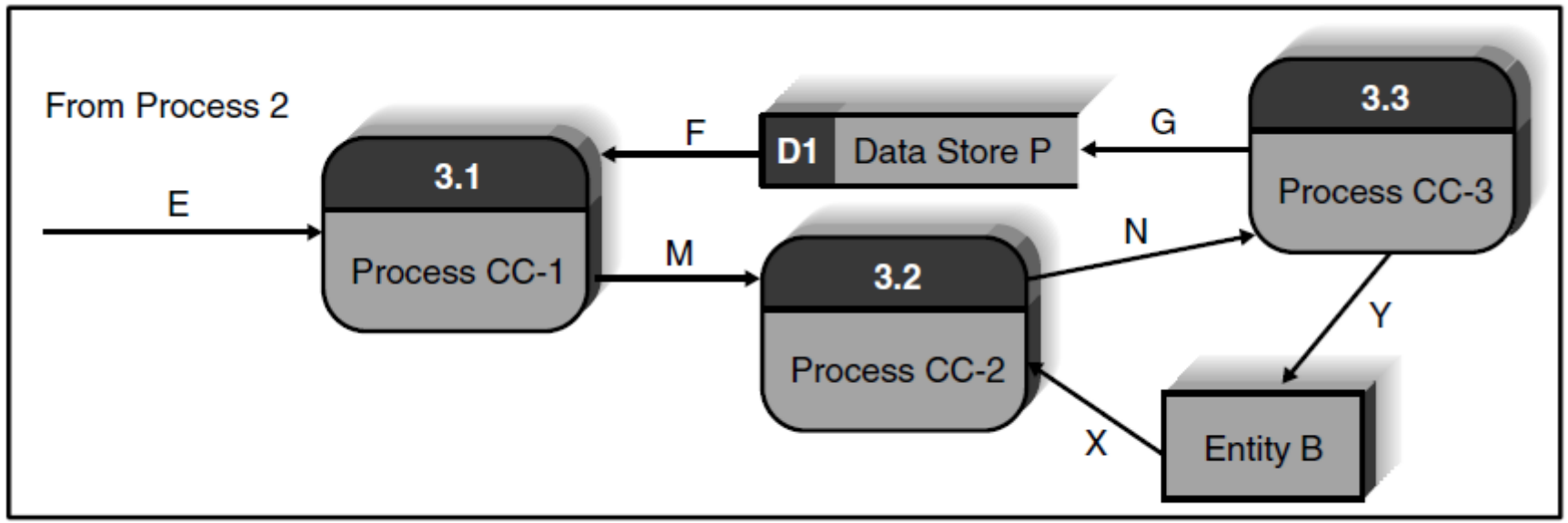
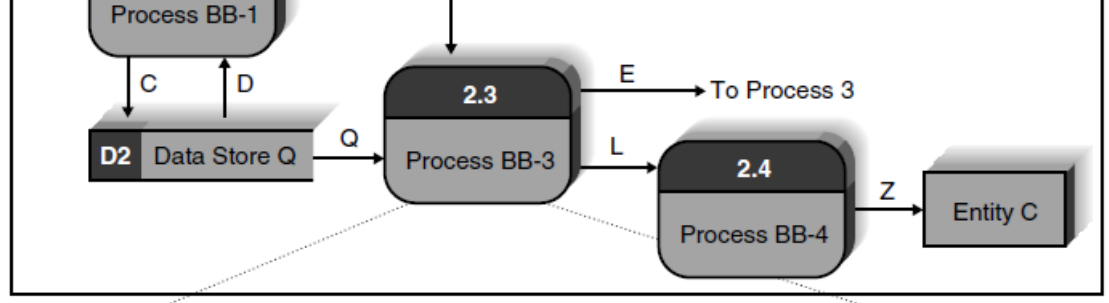


FIGURE 4-15
Processes 1, 2, and 3 level 1 diagrams.



Process 3 Level 1 Diagram

FIGURE 4-15
Processes 1, 2, and 3 level 1 diagrams.



Entry Description

Name: Process BB-3

Entry Type: Process

Process #: 2.3

Short Description: This process calculates Result1 and provides Result1 to process 3, or calculates Result2 and provides Result2 to process 2.4

Process Description:

Receive data flow K from process 2.2
Using data flow K, retrieve correct record (data flow Q) from data store D2
If record meets all selection criteria,
  Compute Result1
  Provide Result1 to process 3 as data flow E
Else
  Compute Result2
  Provide Result2 to process 2.4 as data flow L
EndIf

Notes: Detailed calculations for Result1 and Result2 will be provided by Jeff Essex in the Marketing Dept.

Save | Clear | Copy | Previous | Next | Exit

FIGURE 4-16 Process description entry in data dictionary.

CONCEPTS IN ACTION 4-B | US Army and Marine Corps Battlefield Logistics 1.1.1.1.1.1.1

Shortly after the Gulf War in 1991 (Desert Storm), the US Department of Defense realized that there were significant problems in its battlefield logistics systems that provided supplies to the troops at the division level and below. During the Gulf War, it had proved difficult for army and marine units fighting together to share supplies back and forth because their logistics computer systems would not easily communicate. The goal of the new system was to combine the army and marine corps logistics systems into one system to enable units to share supplies under battlefield conditions.

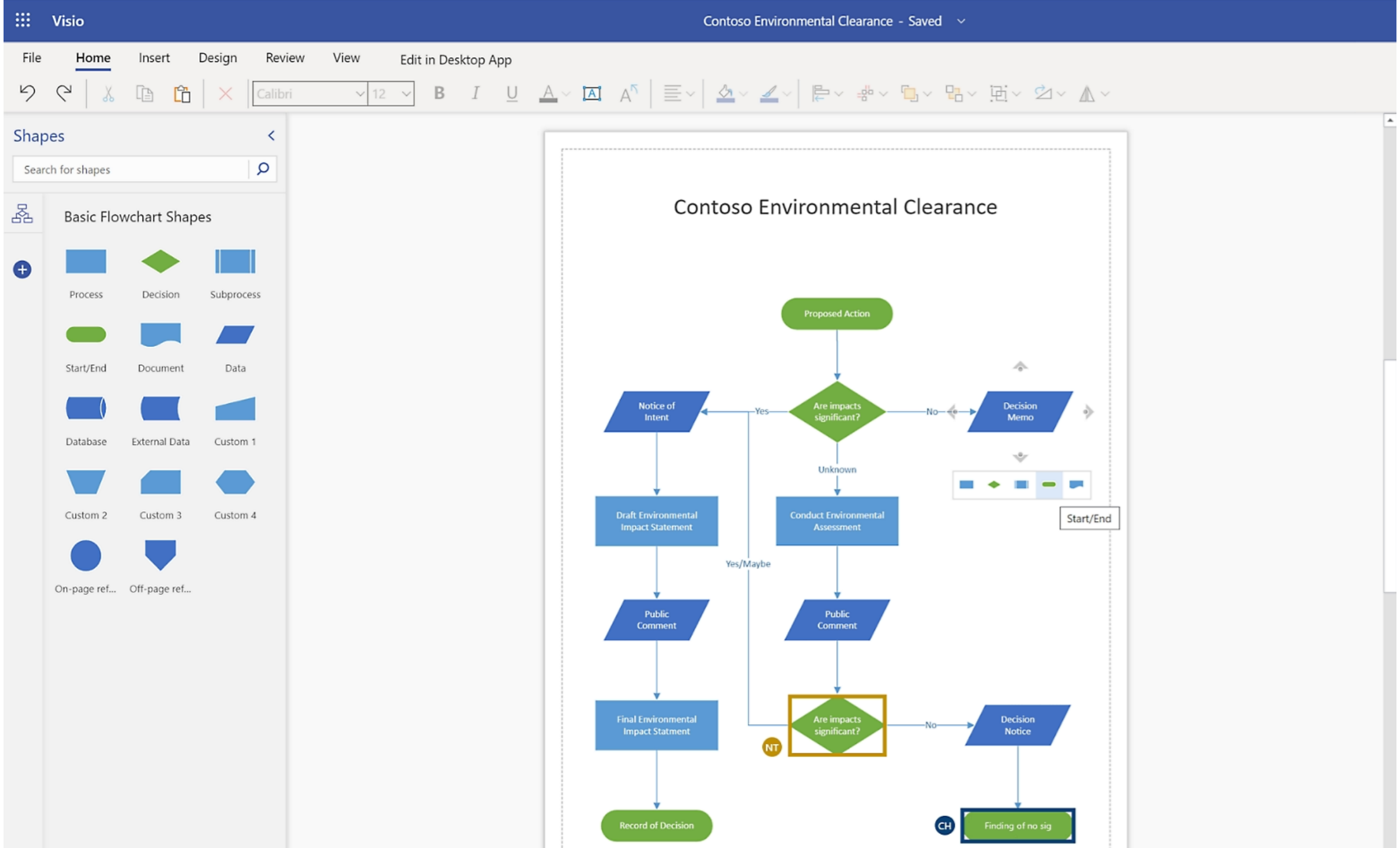
The army and marines built separate as-is process models of their existing logistics systems that had 165 processes for the army system and 76 processes for the marines. Both process models were developed over a 3-month period and cost several million dollars to build, even though they were not intended to be comprehensive.

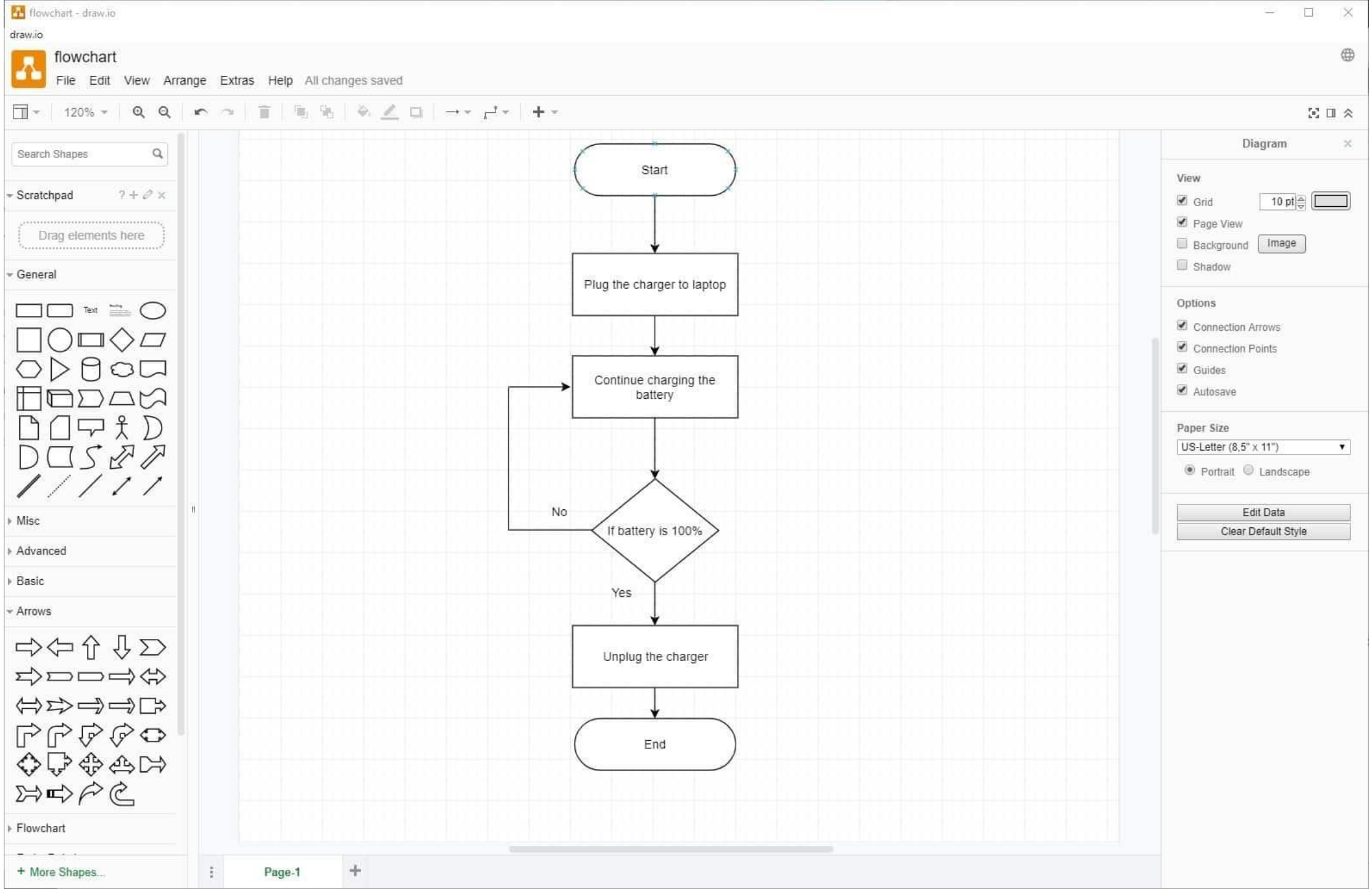
I helped them develop a model for the new integrated battlefield logistics system that would be used by both services (i.e., the to-be model). The initial process model contained 1,500 processes and went down to level 6 DFDs in many places. It took 3,300 pages to print. They realized that this model was too large to be useful. The project leader decided that level 4 DFDs was as far as the model would go, with additional information contained in the process descriptions. This reduced the model to 375 processes (800 pages) and made it far more useful. *Alan Dennis*

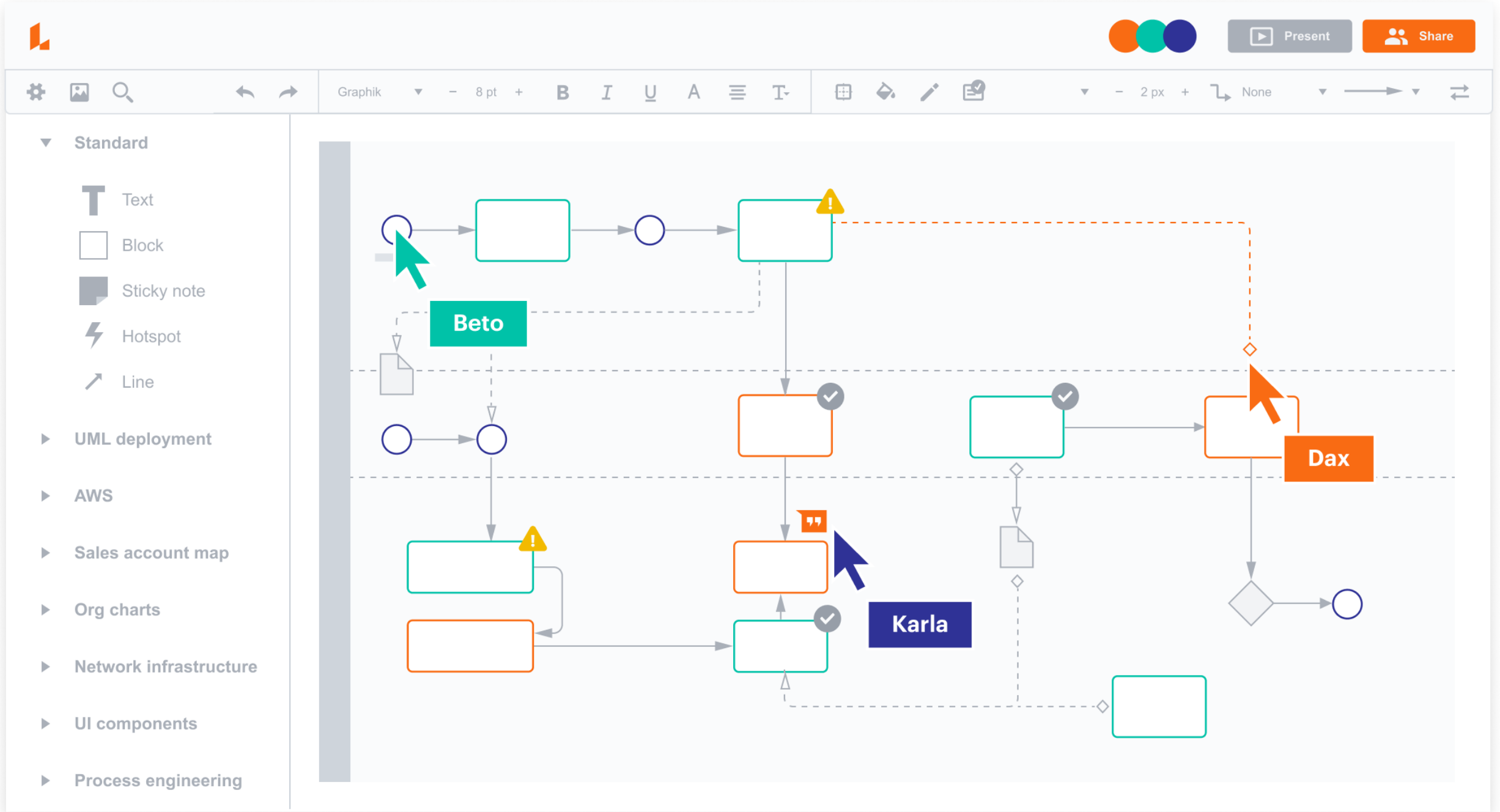
Questions

1. What are the advantages and disadvantages to setting a limit for the maximum depth for a DFD?
2. Is a level 4 DFD an appropriate limit?

หาคนตรวจรับงานยาก ต้องใช้นักวิชาการคอมพิวเตอร์


Visio
Contoso Environmental Clearance - Saved
File
Home
Insert
Design
Review
View
Edit in Desktop App
Shapes
Search for shapes
Basic Flowchart Shapes
Process
Decision
Subprocess
Start/End
Document
Data
Database
External Data
Custom 1
Custom 2
Custom 3
Custom 4
On-page ref...
Off-page ref...
Contoso Environmental Clearance
Proposed Action
Are impacts significant?
Yes
No
Notice of Intent
Decision Memo
Unknown
Draft Environmental Impact Statement
Conduct Environmental Assessment
Start/End
Yes/Maybe
Public Comment
Public Comment
Final Environmental Impact Statment
Are impacts significant?
NT
No
Decision Notice
Record of Decision
CH
Finding of no sig


flowchart - draw.io
draw.io
flowchart
File Edit View Arrange Extras Help All changes saved
120%
Search Shapes
Scratchpad
Drag elements here
General
Misc
Advanced
Basic
Arrows
Flowchart
+ More Shapes...
Start
Plug the charger to laptop
Continue charging the battery
If battery is 100%
No
Yes
Unplug the charger
End
Page-1
Diagram
View
Grid
10 pt
Page View
Background
Image
Shadow
Options
Connection Arrows
Connection Points
Guides
Autosave
Paper Size
US-Letter (8,5" x 11")
Portrait
Landscape
Edit Data
Clear Default Style


Present
Share
Graphik
8 pt
2 px
None
Standard
Text
Block
Sticky note
Hotspot
Line
UML deployment
AWS
Sales account map
Org charts
Network infrastructure
UI components
Process engineering
Beto
Dax
Karla