CHAPTER 4

Process Mapping II

Introduction

This chapter continues the discussion of process map variations and different methods of depicting aspects of processes. In addition, many maps are so complex that they are difficult to understand unless they are summarized in logical "chunks." This chapter explains the reasoning for creating summary maps. Once the variations have been discussed, we move to validating and verifying a process map.

Novices make errors in process maps that slow their learning of the techniques. Therefore, several common errors are discussed along with correct mapping techniques.

Process Map Variations

Process maps may contain many variations, including the following:

- multiple inputs or outputs to a single process
- multiple conditions
- use of connectors
- multiple swimlanes participating in a process step
- handoffs
- loosely coupled processes
- multiple methods of accomplishing a process step
- manual data processing
- time-driven processes

Each of these variations is discussed in this section. Please keep in mind that all examples in this section show partial process maps and hence may have no "start" or "stop" icons or swimlanes.

Multiple Inputs or Outputs to a Process

Frequently, a process will be preceded by more than one parallel activity. For instance, an approval on a document may have parallel client reviews that converge to a joint "approve project" process. Figure 4.1 shows the form of such a figure.

Similarly, a map may have multiple following-process steps, as in Figure 4.2. An example of such an outcome might be "approve project," which spawns both a printing activity and an activity to put the project documents on an intranet.

Multiple Conditions

Another situation that frequently arises is one in which there are multiple outcomes from a single decision. This discussion recaps an example from chapter 3. The constructions are determined by the selection of a default.

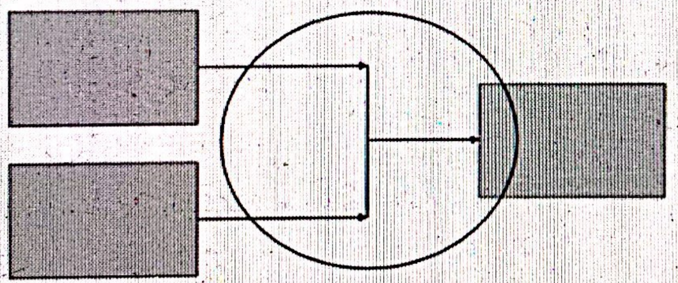


Figure 4.1. Simultaneous predecessor processes.

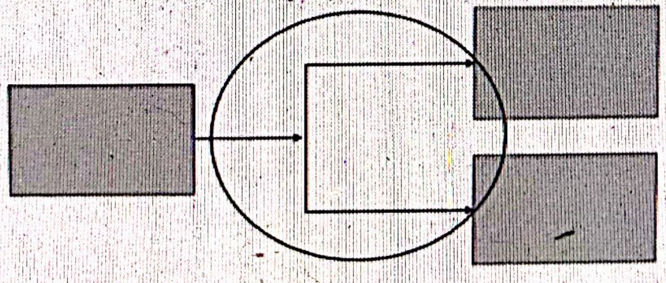


Figure 4.2. Simultaneous outcomes.

A *default* is a process step that is executed when none of the options explicitly checked is chosen.

Multiple decision constructions have two alternative forms depending on whether or not there is default processing. The construction shown in Figure 4.3 has a default error process if none of the four alternatives specifically tested is selected.

Figure 4.4 shows the fourth subprocess as the default. In this construction, there is no error processing, and, if options one, two, or three are not selected, then the fourth is automatically selected.

Another example might be months of the year. There would be 11 decision diamonds, with the 12th month as default if there is no error processing. Conversely, if there was error processing or some "other" process where none of the options is selected, then there would be 12 decision diamonds with the "other" option as the default.

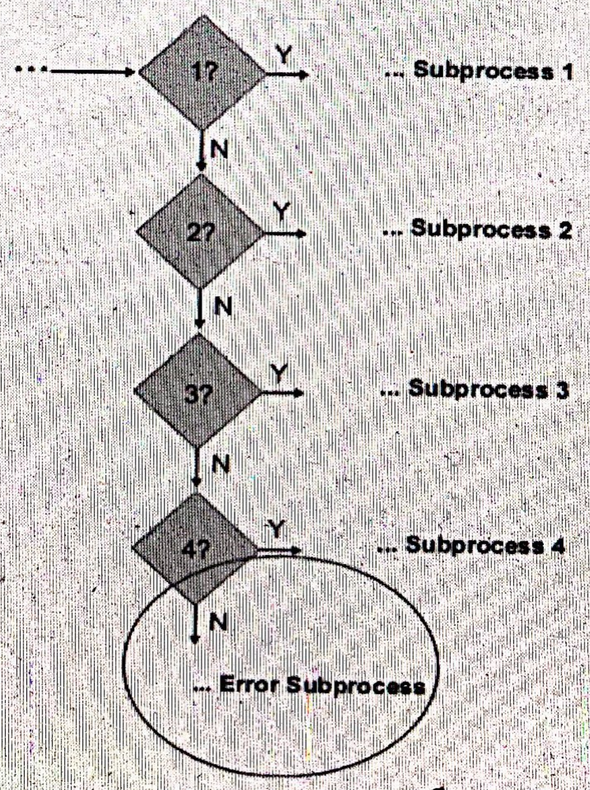


Figure 4.3. One subprocess with four alternative actions, each with a "no" process.

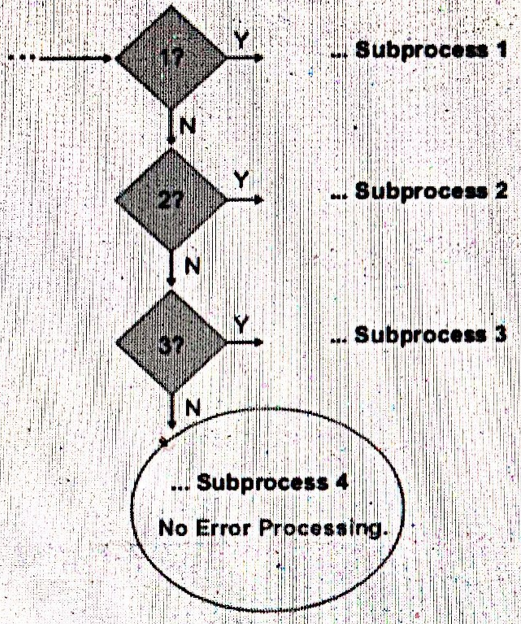


Figure 4.4. One subprocess—four alternative actions.

Use of Connectors

Recall that connectors are small circles that are used to improve the readability of a diagram. Figure 4.5 repeats the definition of a connector.

Connectors are the source of many diagram errors because they are frequently misused. As Figures 4.6 and 4.7 demonstrate, a connector is created as an output of a process and reused to show input to the next sequential process step. Figure 4.6 shows connector "1" as created to change the flow of the process.

Figure 4.7 shows connector "1" as indicating that the process flow continues at the process step to which it points. Each number set is used once except, as discussed, in the case of using single connectors from multiple locations that stop a process.

Connectors are used in a number of situations to improve diagram readability. Connectors can be used to indicate a single process stop point, manage flow of process details, or manage complex diagrams so lines do not cross. In general, connector numbers should be sequentially assigned. In the case of having a single "stop" icon, many process steps may have a connector that goes to a single stop. All of the process steps going to a single stop should have the same number.

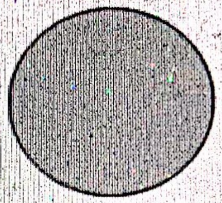


Figure 4.5. A connector, shaped like a small circle, is used to show a change of process flow and to connect to the point where the flow continues.

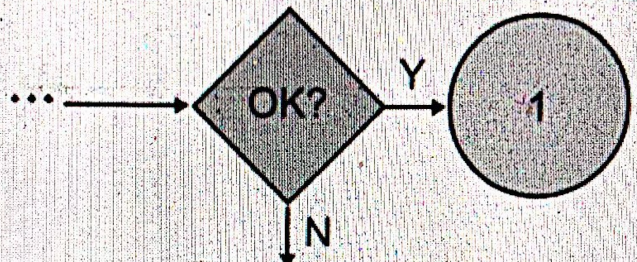


Figure 4.6. Connector change of process flow.

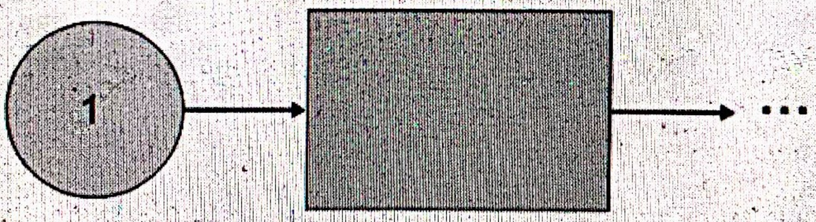


Figure 4.7. Connector connects where flow continues.

Another reason for using connectors is to simplify complex diagrams. Connector use increases readability, especially in situations where a set of decisions has many possible outcomes. Figure 4.8 shows such a situation. The important item here is the set of decisions and their criteria. The lesser important items are the connected-to logic, showing what happens in each situation.

Figure 4.9 shows a diagram for which connectors could be used to avoid overlapping lines and complex logic. When lines cross, go backward rather than forward, or span many swimlanes, readers tend to get lost, and diagrams become unreadable. By using connectors to untangle

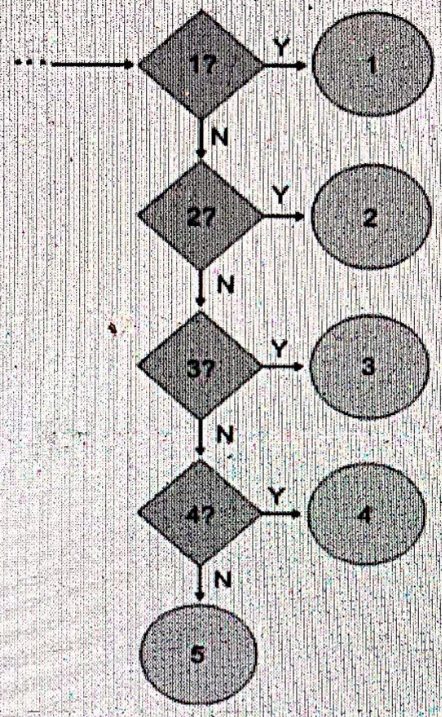


Figure 4.8. Multiple connectors—a variety of reasons.

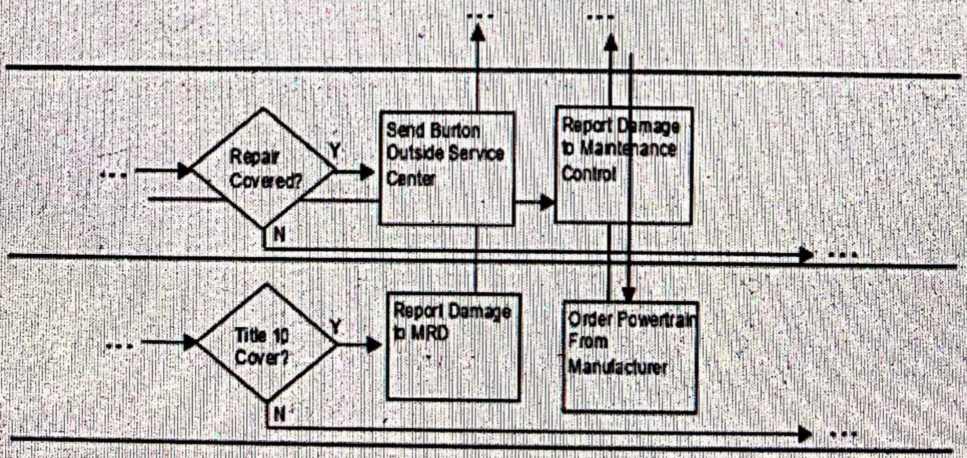


Figure 4.9. Connectors could enhance readability.

the lines, and by keeping the arrows short and of consistent length, readability is enhanced.

Finally, Figure 4.10 shows a simple example of connectors used to identify multiple stop points in a process. Imagine a more complex diagram with 10 possible stop points. Using ten connectors all number "99" to connect to a single stop point clarifies the diagram.

Another convention is to use a number that would otherwise not occur to signal a "stop" connector, such as 99. As more process maps are created, use of a single "stop number" provides a visual cue that facilitates user understanding of the maps.

Multiple Swimlanes in a Process

For meetings, interviews, group decisions, or other types of multiperson activities, multiple swimlanes may participate in a single process. Figure 4.11 shows a larger box enclosing multiple swimlanes that participate in a single process.

When the swimlanes are adjacent, the process box is drawn once, with one set of words in the box, centered to identify that both parties take place in the process. If more than two swimlanes participate in the process, and the swimlanes are adjacent, one process box would span all required lanes.

Process-step actions taken may differ by participant, in which case multiple process boxes can be placed within the multilane box to show the details. Actions requiring multiple swimlane participation include

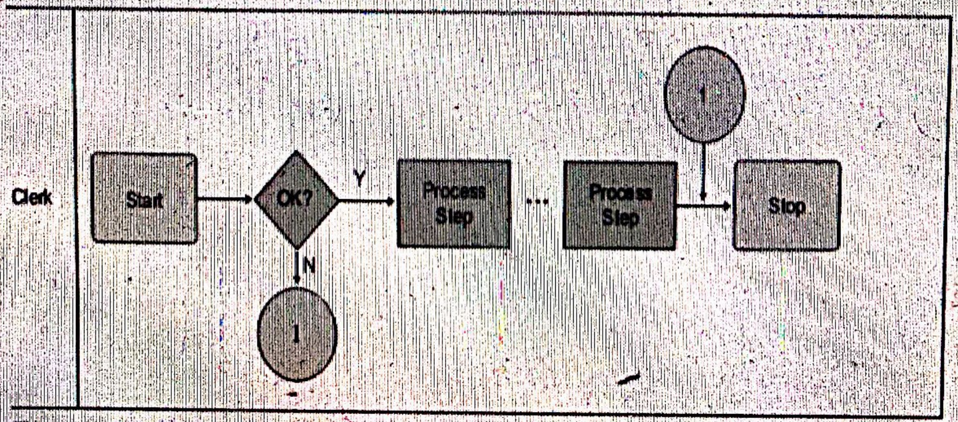


Figure 4.10. Connectors manage stop points.

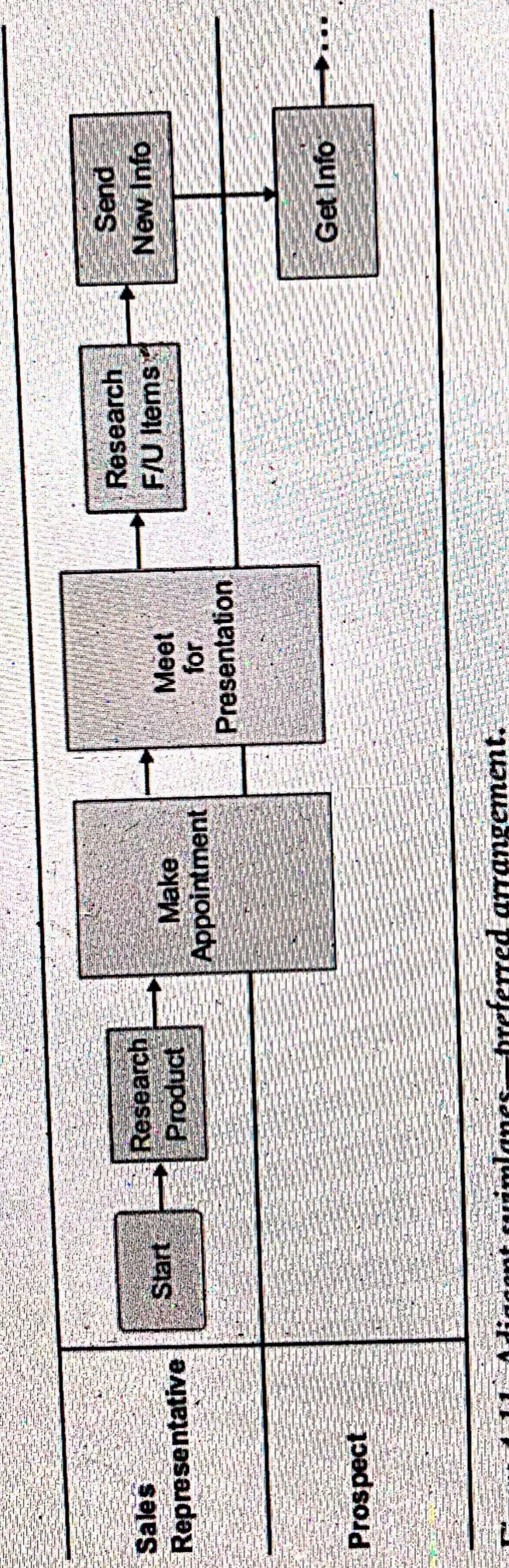


Figure 4.11. Adjacent swimlanes—preferred arrangement.

phone conversations and meetings. Both of these may have participants discussing a situation, with one of the participants making a decision. In such a situation, multiple process boxes would be in the multilane process box.

An example is the "present product" and "examine product" process step in Figure 4.12, where the sales representative is making a sales call. Both swimlanes participate in the process, so a large, shaded box is drawn, but each swimlane has a different step, each in its own process box.

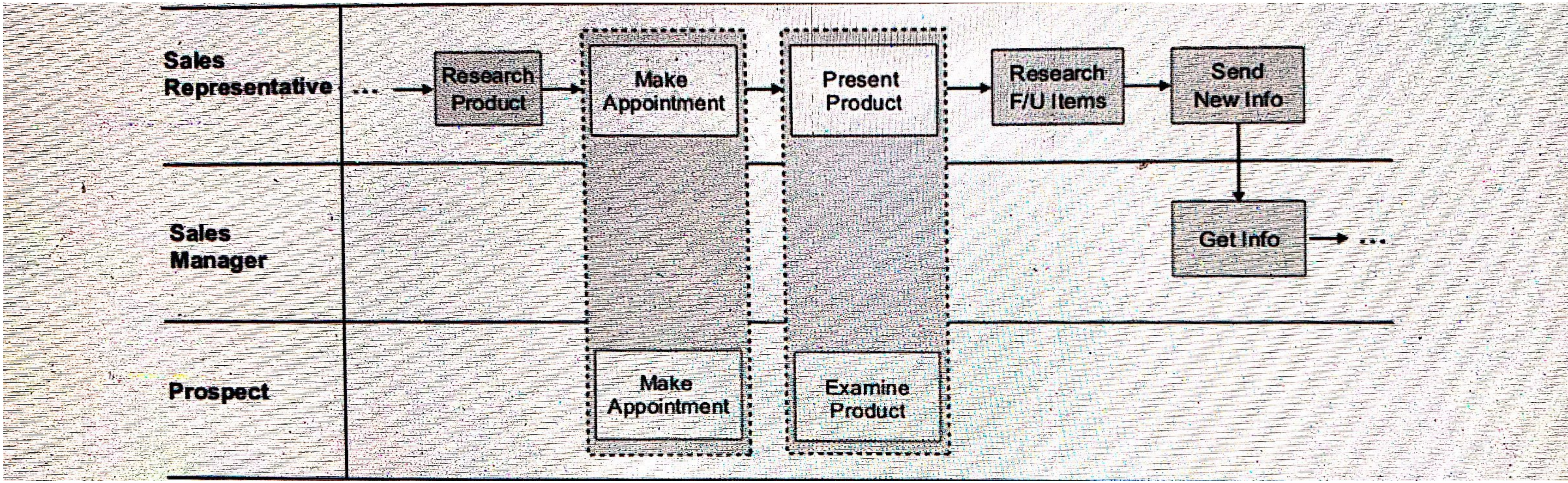
Since swimlane diagrams should be read from top to bottom and from left to right, swimlanes may participate in a single process step but may not be adjacent. Figure 4.12 also depicts this situation by the drawing of one large box—which can be a different shadow color or have a dotted line when no colors are used—to show that multiple swimlanes are involved in the process step. Then, for the participating swimlanes, a process box is drawn, with the same words in each of the process boxes, or different words if the actions taken during the joint activity differ.

Handoffs

Using a diagram similar to a previous one, appointments might be stored in some type of software. The processing for software handoffs is shown in Figure 4.13.

Each side of a handoff has a process step for each handoff that matches the real-life interaction. A sales representative, for instance, enters contact information into an application and presses "enter." The application takes the screen information, processes it, and confirms the process by displaying some type of acknowledgement. The sales rep views the display that confirms the outcome of the data entry.

Handoffs occur between people, between people and applications, or between people and roles (i.e., a person with a nonspecific title, such as "customer"). Handoffs are a frequent source of error both in process maps and in the actual process because they tend to be taken for granted, which allows the information transferred to be lost. As a result, when mapping a process for which the sources of errors are unknown, it is important to include all handoff details in order to ensure complete analysis.


Sales Representative
Sales Manager
Prospect
Research Product
Make Appointment
Present Product
Research F/U Items
Send New Info
Get Info
Make Appointment
Examine Product

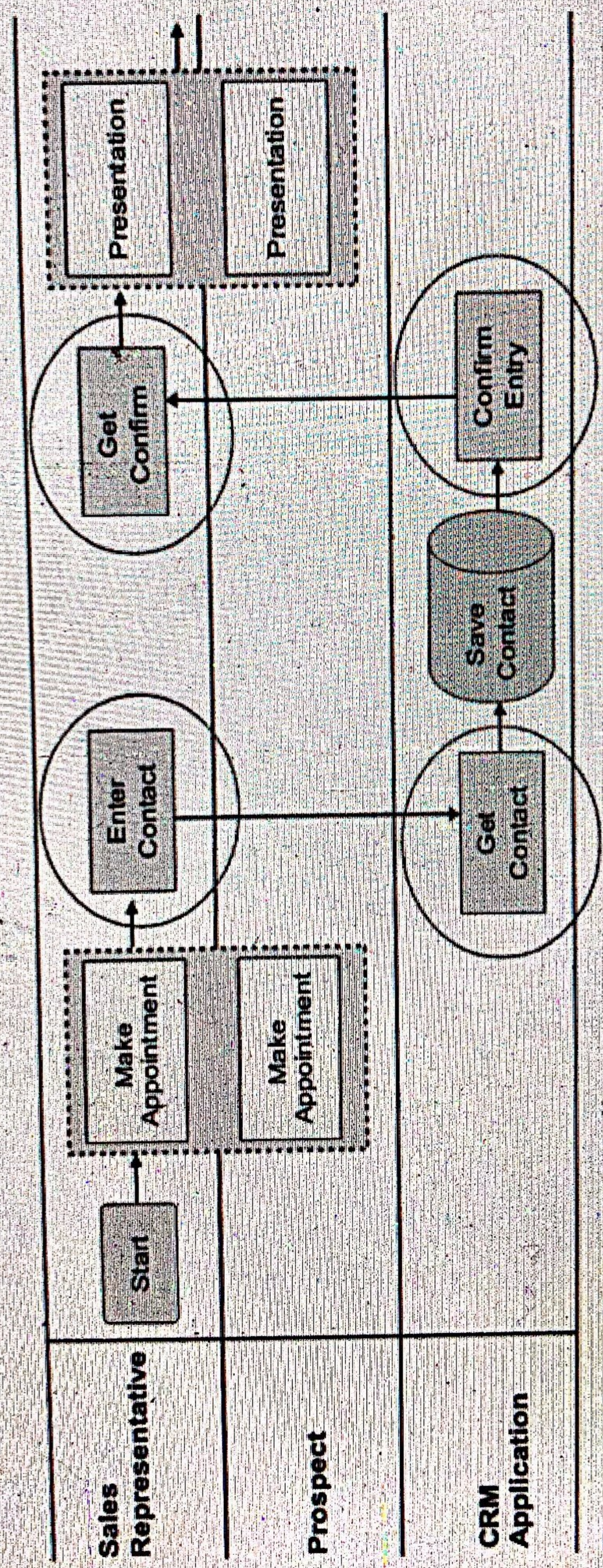


Figure 4.13. Application handoffs.

Loosely Coupled Processes

A *loosely coupled process* is one that contains discontinuous processes with long or planned delays, no direct interaction, and no necessary impact on one part of the discontinuity when changes are made to the other part of the discontinuity. Therefore, loosely coupled subprocesses, when embedded in complex processes, can be broken out as their own processes. An example of this situation frequently occurs in business when a request is entered on a website or intranet site and, at some specified time, someone checks for requests and handles them. Examples are customer-order entry and help-desk requests.

Loosely coupled processes could be handled as a single process, with a dotted arrow showing the discontinuous wait time, or as two processes. If the process is simple, then there may be nothing served by separating it into two processes. If the process is complex, however, then separating the subprocesses can enhance understandability. Examples of split subprocesses are shown in Figures 4.14 and 4.15. Figure 4.14 shows an employee entering a request via an intranet, receiving a confirmation of request completion, and ending the activity. Figure 4.15 depicts a human resources (HR) employee getting requests, verifying request completeness, and either rejecting or accepting the request and then updating the request status and continuing with other processing. This is a complex process that does not involve direct interaction between HR and the employee; therefore, there is no real need for the employee subprocess to be shown with the HR subprocess. The intranet links the employee to HR. Since interaction is through an intranet, the subprocesses can be split, and each subprocess becomes its own process.

The risk of splitting a large process into two subprocess diagrams is that a delay between the two parts (in this case, request entry and request processing by HR) could be overlooked. Therefore, when decoupling subprocesses, care must be taken so that the linkage delay is not forgotten during process analysis.

Multiple Methods of Accomplishing the Same Work

Another variation occurs when a swimlane is defined as an organization or a department, such as HR, where different people execute the

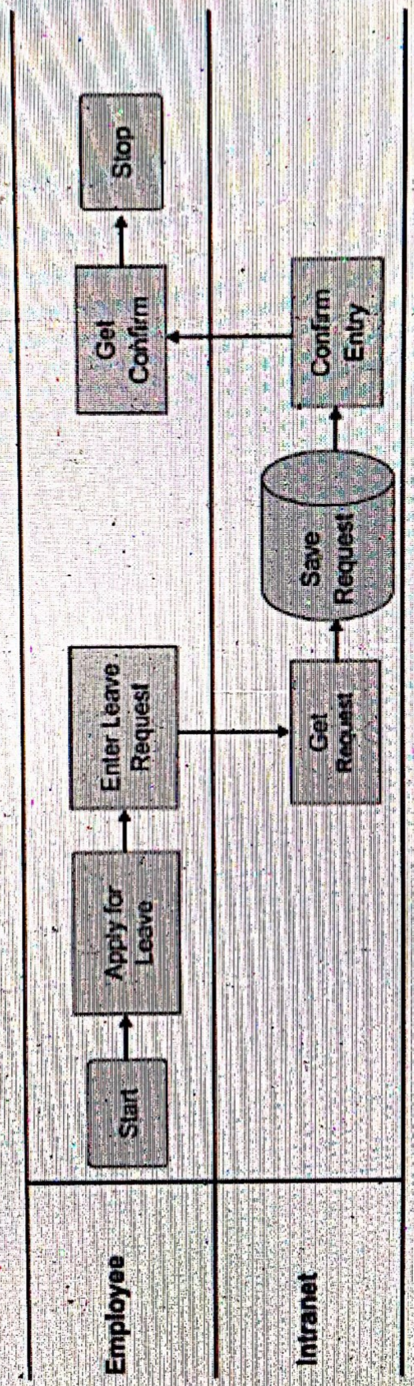


Figure 4.14. Part 1 of discontinuous processes = multiple processes.

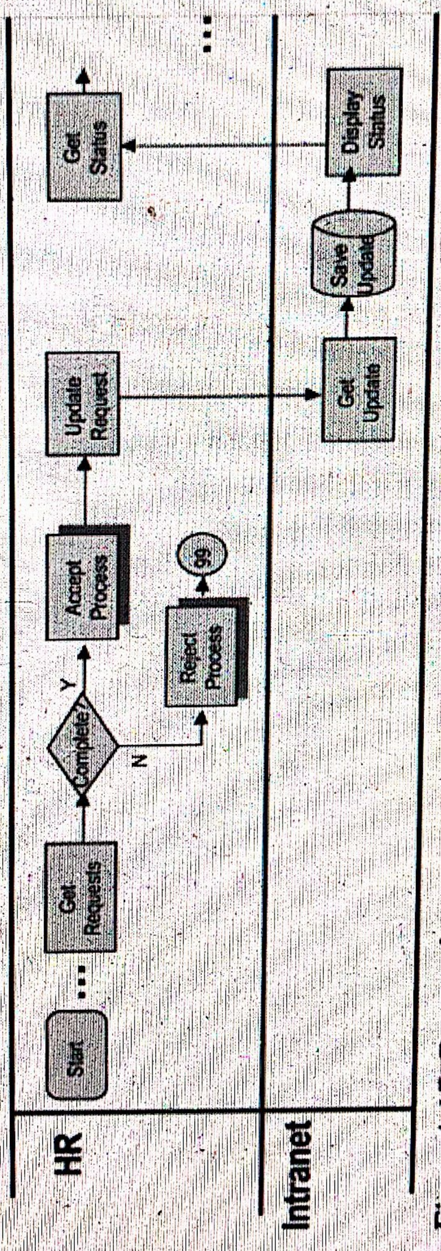


Figure 4.15. Part 2 of discontinuous processes = multiple processes.

process differently. There are four ways of depicting this situation, and the reasoning behind which one is appropriate depends on the context. The four methods include the following:

- separate swimlanes
- condition within a swimlane to show differences
- subswimlanes
- different processes

Separate Swimlanes

These techniques show several ways to depict the same work. The first method, separate swimlanes, has two variations that both identify the organization in some way but draw wholly separate swimlanes. This variation is shown in Figure 4.16. This method would be selected if there were many points of divergence or the processing of one or both of the sublanes is complex.

Condition Within a Swimlane

Trivial divergences can use a condition within a swimlane via a condition diamond, as shown in Figure 4.17. The conditional treatment shows one of the possible people (clerk) performing the subprocess. The disadvantage of this method is that the reader is left guessing who the other person is. The manager could be mentioned rather than using "no" as a leg of the process, but that naming convention is not the best.

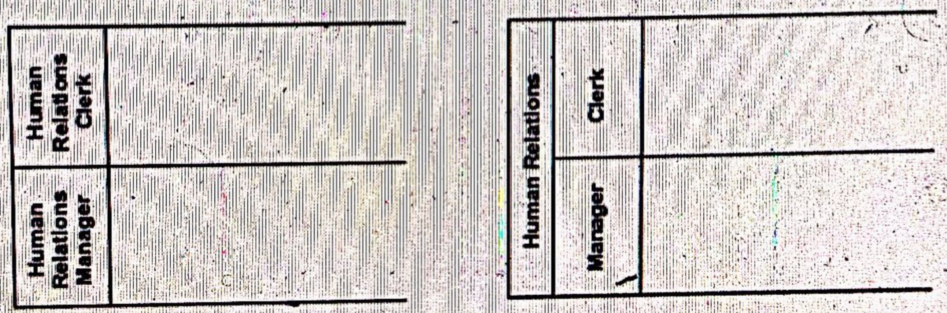
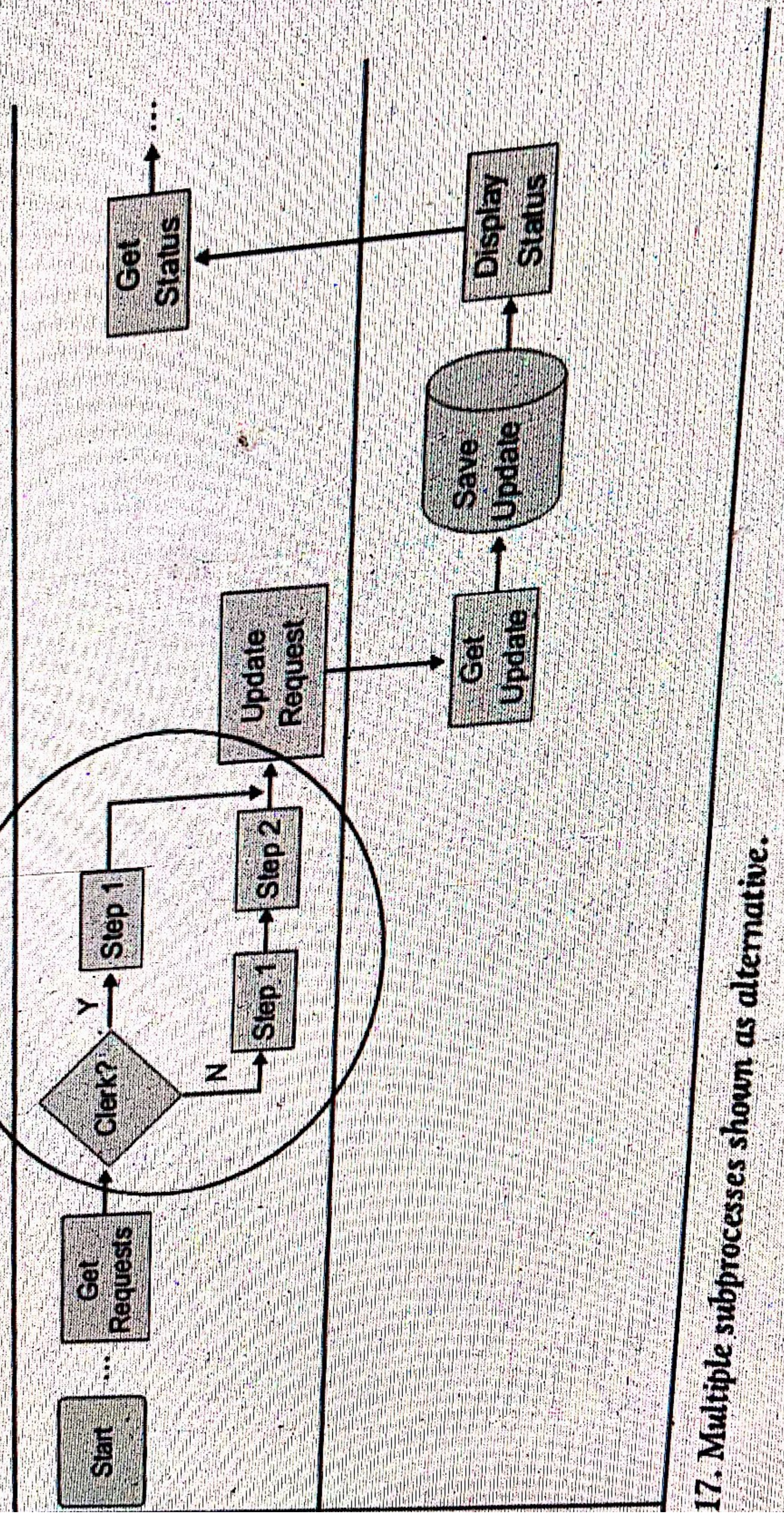


Figure 4.16. Multiple subprocesses shown as swimlane alternatives.



17. Multiple subprocesses shown as alternative.

A variation that would remove this disadvantage would be to include a condition to check for "whom"—one leg would identify "clerk," while the other leg would identify "manager" rather than showing "yes" or "no" alternatives. The advantages of this method are brevity and improved clarity of action.

Subswimlanes

The preferred method for minor divergences is to embed subswimlanes at the point at which they diverge. Figure 4.18 shows the condition leading to either a manager or clerk sublane.

Different Processes

A mapping exercise to make processes consistent might find a merged or acquired organization with its own process. Trivial differences should be mapped using one of the previously mentioned techniques. If there are significant differences, each organization's process should be mapped completely and separately, and then each process should be compared to the acquiring (or desired) organization's process to identify and reconcile the differences.

Manual Data in Processes

Manual data offer another complication to be dealt with in diagrams. One method is to use a bidirectional arrow between the paper document and the person using it. The other method is to show the flow of the process continuing from the document to the next process. A bidirectional arrow is preferred, since a paper document cannot, by itself, further a process.

A document's movement or content might alter what a person or application does, but the interactions with the document take place before the subprocess step.

When a document's content can alter or move a process forward, it is acceptable to use an arrow from the document to show its significance (see Figure 4.19). However, this subtlety is missed by most readers, and therefore, the bidirectional arrow is more commonly used. Companies develop their own local standards for process mapping, and this is one of the areas needing a decision on how it will be handled.

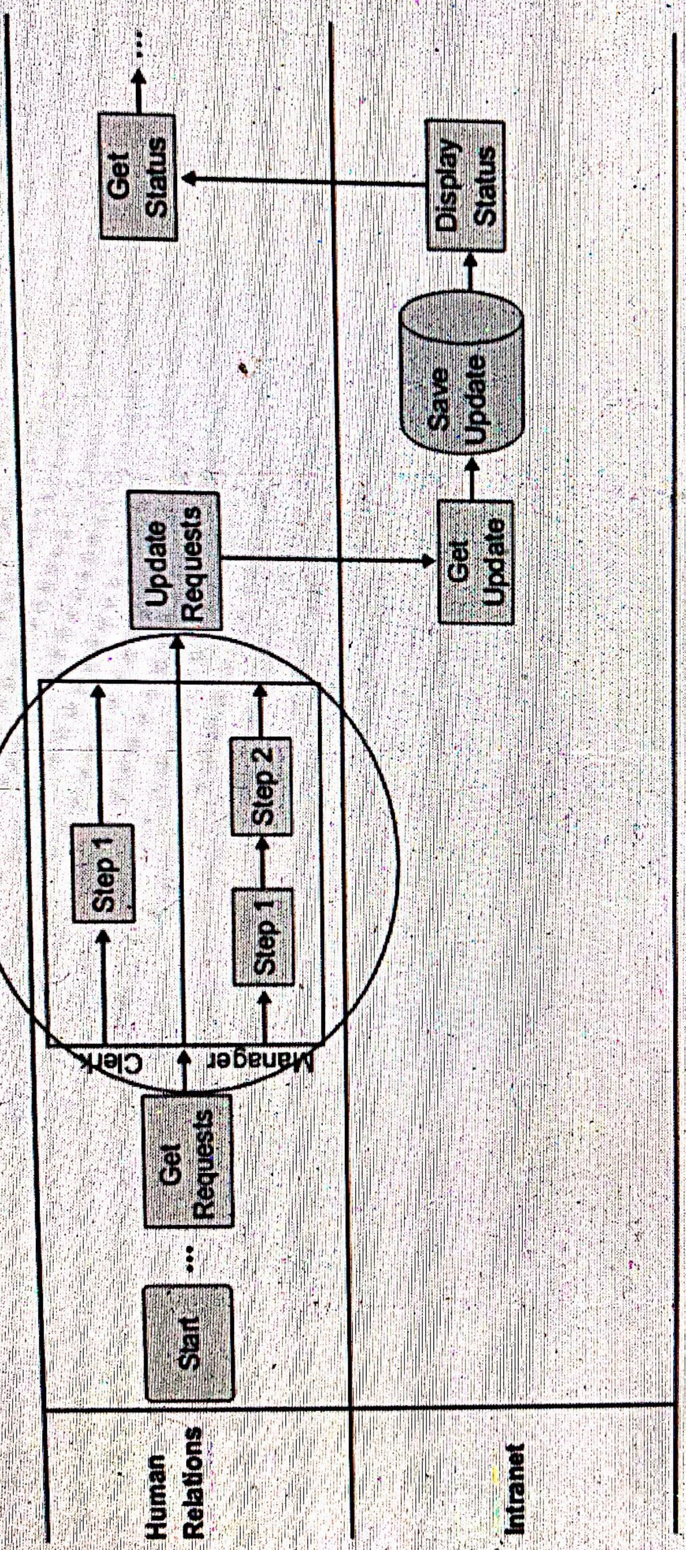
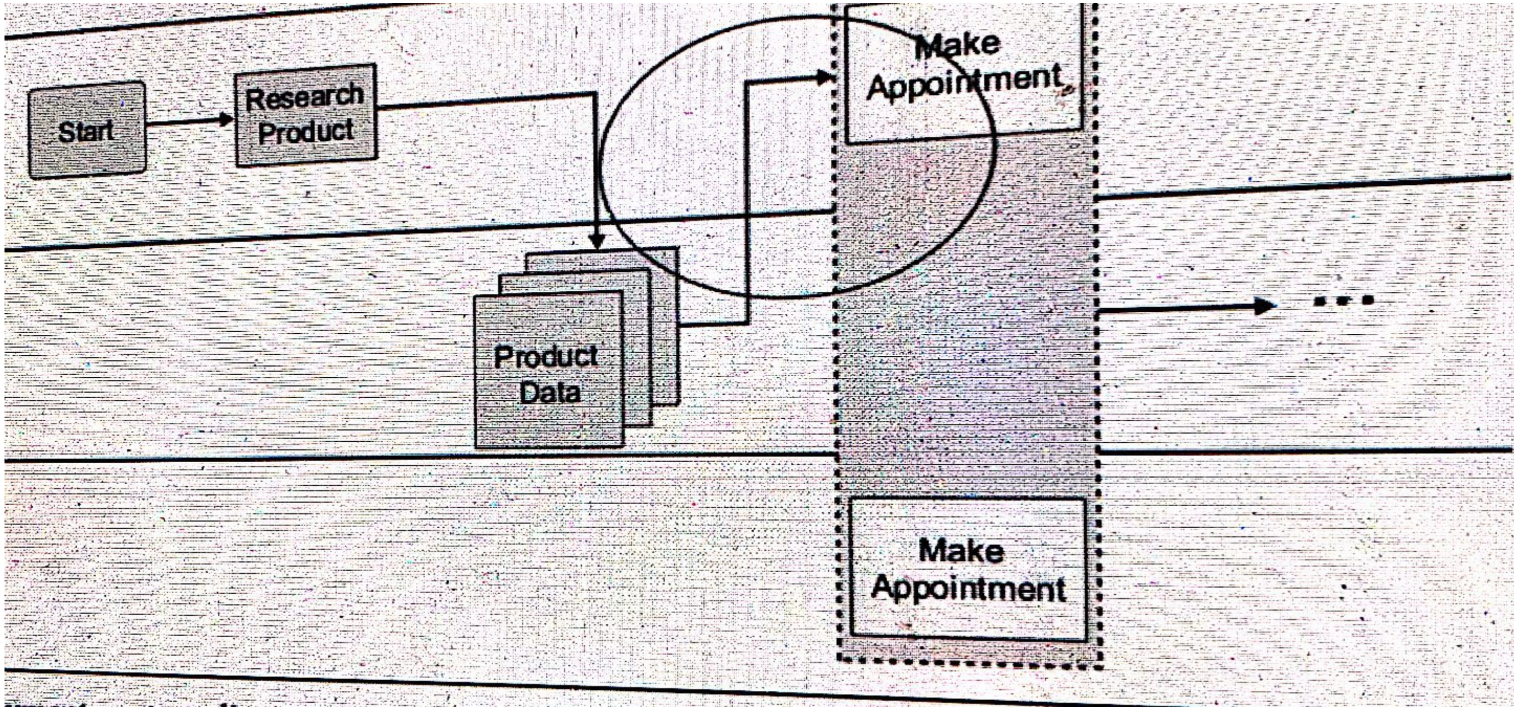


Figure 4.18. Multiple subprocesses shown as subswimlanes.


Start
Research Product
Product Data
Make Appointment
Make Appointment

An alternative to this process depiction is to show the entire Internet interaction from which product data are retrieved. Depending on the goal of the mapping exercise, showing the interactions may be crucial to defining improvements and would therefore be the desired mapping approach.

Time-Driven Processes

Another variation in process maps is the depiction of time-, geography-, or event-driven process steps. Figure 4.20 shows a time-driven process example for a church service during which visitor cards may be available on the seats. If a card is present, and the swimlane is for a new attendee, then the person must decide whether or not to put the card in an offering basket. The time-driven aspects of the process are the start and end of the service. There might also be process steps for collection of money or other aspects of the service. Another example might be a time-driven process that has no specific end date, such as university graduation, obtaining a patent, consolidating organizations, and so on. In these examples, the end date is time zero or t_0. All other dates are some number of days, such as 10, which would be t_{10} from the event date. The times then act as a count down to the t_0 event date.

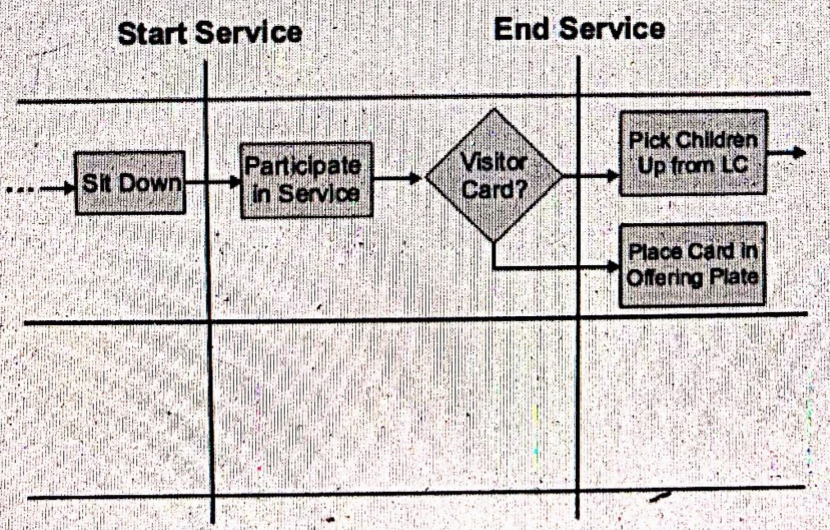


Figure 4.20. Time-driven diagram example.

Summary Process Maps

Frequently, process maps will be complex, span multiple pages, and require their own "map." When a team is developing process maps, it becomes clear that summaries are needed. Usually, a junior or part-time team member will say something like, "I wish I had a map to the maps!" or "I've lost my train of thought on this process." Comments such as these are clues that summaries are needed. Summary maps are created as an aid to understanding. In this section, we develop a process for creating summary maps and discuss how best to automate them.

Creating a Summary Map

There are two basic methods for developing map sets: bottom-up and top-down. Neither is preferred, but each person will probably have a preference—relating to how he or she tends to think—since most people are more comfortable with one or the other method. It is important that whichever method is chosen, the maps are numbered so that connectors are consistent within the set and so that when a number is used, its meaning is consistent within the set of maps. Bottom-up mapping is discussed first.

Bottom-Up Process Analysis

A bottom-up process for creating summary maps is as follows:

1. Create the process map(s).
2. Identify subprocesses on the original map.
3. Convert the subprocesses to summary icons.
4. As needed, repeat steps 1–3 until you have a one-page overview map.
5. Create "drill-down" maps for each subprocess.

The advantage of bottom-up development is that complexities and details that might alter a complete view of the process are known early on. The disadvantages of bottom-up development are that the details may obscure the recognition of the subprocesses and the analysis may result in an incomplete or skewed view of the overall process. Summary maps help overcome the risks possible with bottom-up process map development.

Top-Down Process Analysis

Conversely, some people think of the major process steps first and then follow to get the details. This is a frequent strategy in the initial interview in defining the scope of a process. In this case, the process of creating diagrams is top-down, and the steps are as follows:

1. Identify the major subprocesses.
2. Create and validate a summary process map.
3. For each subprocess on the summary map, do the following:
   a. Research, interview, or observe to obtain the process details.
   b. Create a process map for each subprocess.

The advantage of top-down process analysis is that all major steps are identified first so that completeness is verified and details can be deferred until the major steps are known. Two disadvantages of top-down process analysis are that the complexities affecting the time to complete analysis may not be known until late in the data-gathering activity and that details altering the outcome of the analysis may be skipped altogether.

Analyze your own thinking processes as you progress through the book to determine which type of thinking you are most comfortable with, and then follow that method on any projects you develop. Some may find that a combination of top-down and bottom-up will work best. Obtain as many details as possible to develop a summary diagram, and then switch to bottom-up thinking to develop the individual detailed maps.

Compiling a Diagram Set

Complex processes may have multiple levels of diagrams developed to allow discussions with executives (least detailed), people managing the work (more detailed), and people who actually do the work and systems developers (most detailed). Regardless of the number of levels in a diagram set, all stakeholders should review a set of the maps, each one with an appropriate level of detail. A final set of process maps should include the summary map and all related detail or "drill-down" maps (see Figure 4.21).

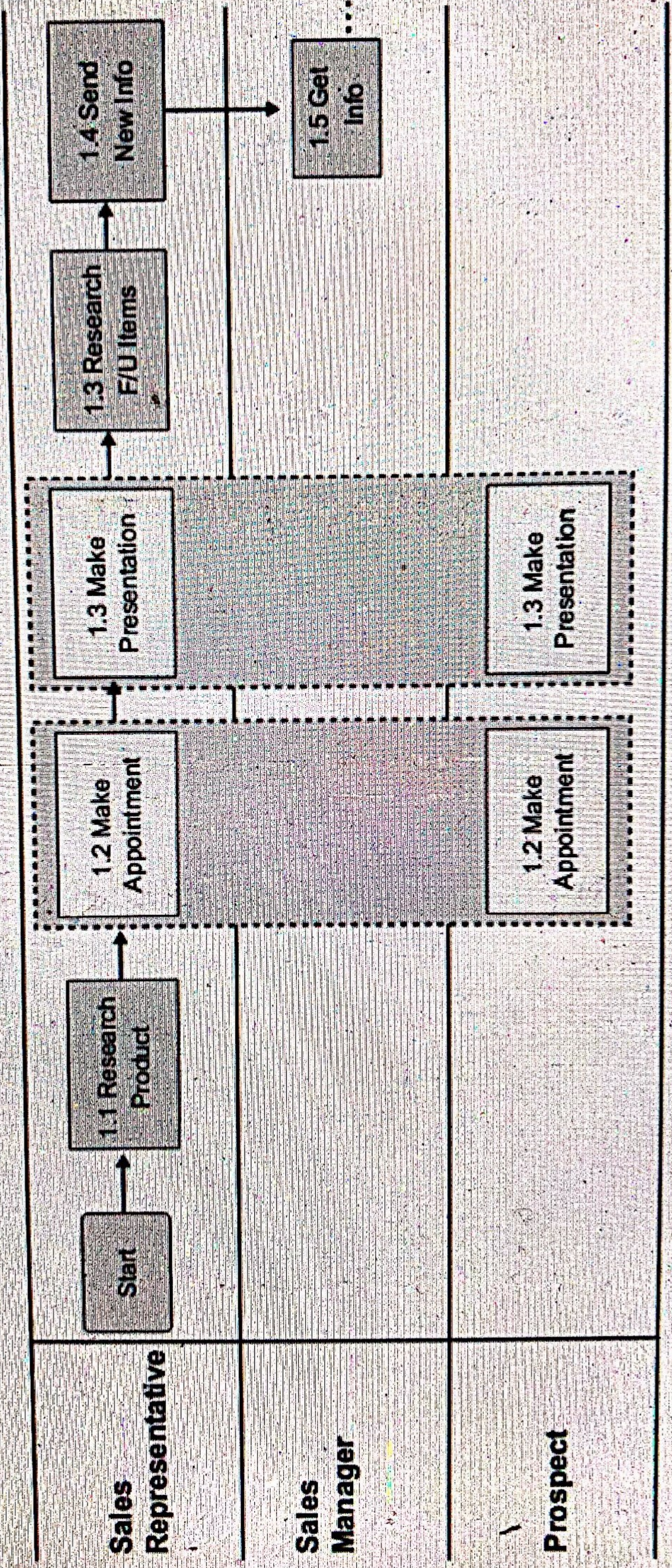
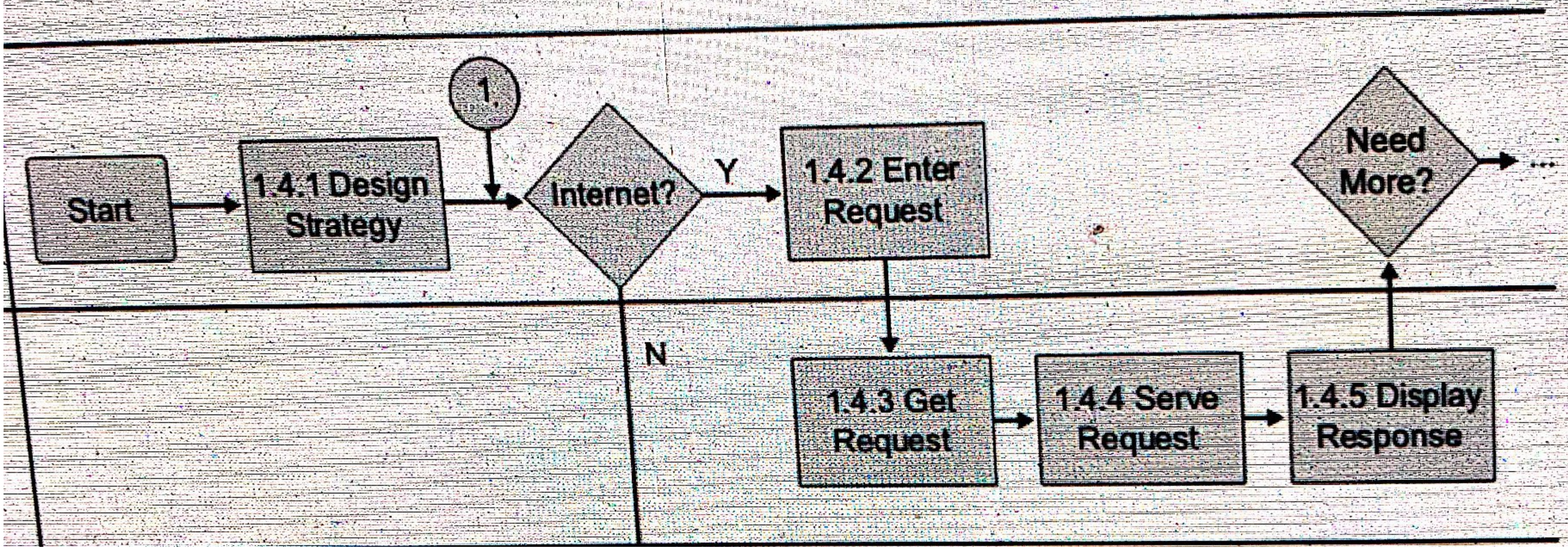
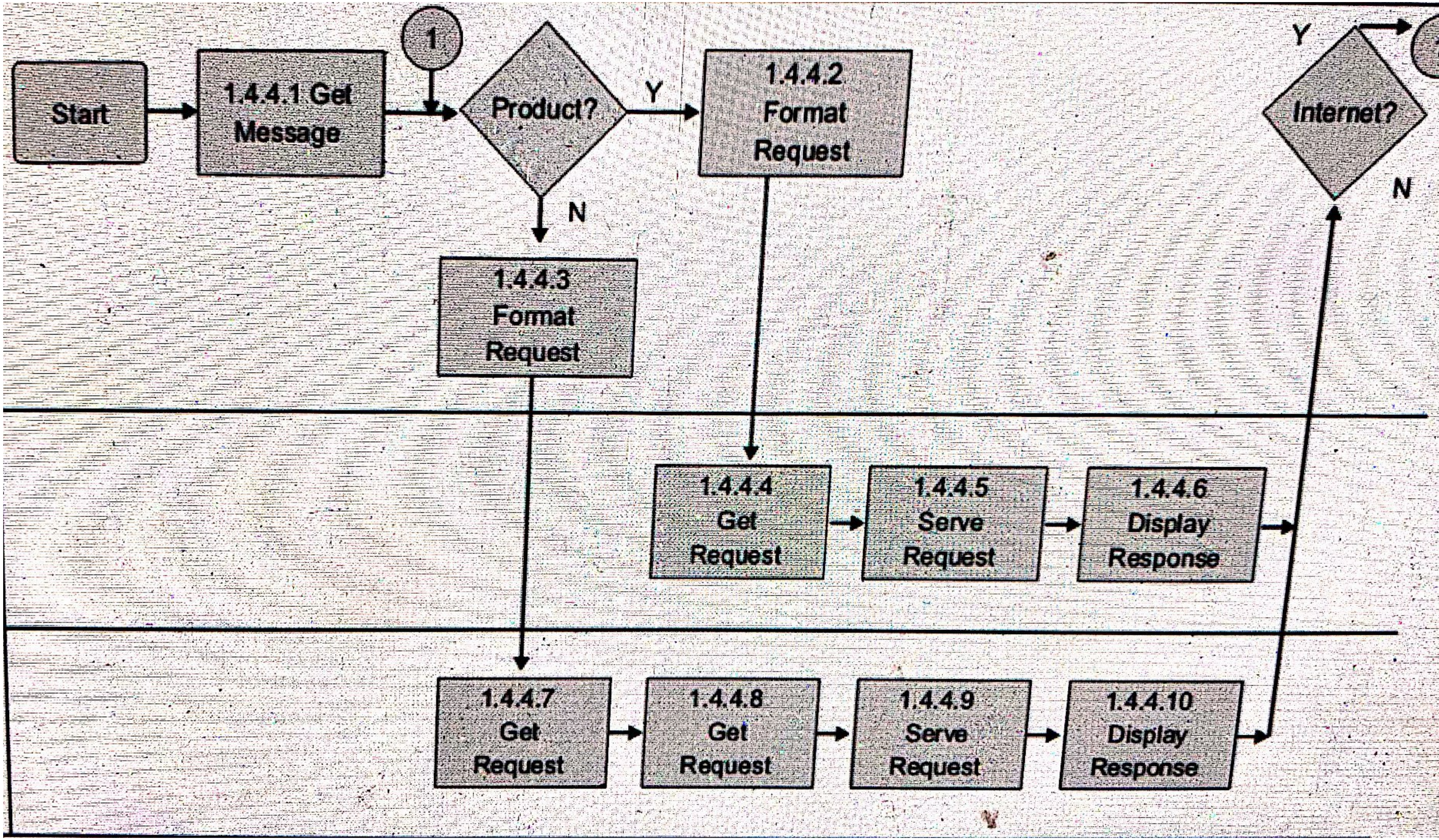


Figure 4.21a. Process map sets.


Start
1.4.1 Design Strategy
1.
Internet?
Y
N
1.4.2 Enter Request
1.4.3 Get Request
1.4.4 Serve Request
1.4.5 Display Response
Need More?


Start
1.4.4.1 Get Message
1
Product?
Y
N
1.4.4.2 Format Request
1.4.4.3 Format Request
1.4.4.4 Get Request
1.4.4.5 Serve Request
1.4.4.6 Display Response
1.4.4.7 Get Request
1.4.4.8 Get Request
1.4.4.9 Serve Request
1.4.4.10 Display Response
Internet?
Y
N
1

Each level of map should be numbered starting at either zero or one for the highest level and moving down through the hierarchy in a logical system, such as what follows:

1. Level 1—Summary map
   1.1. Level 1, Process box 1
      1.1.1. Level 1, Process 1 detail diagram process box 1
      1.1.2. Level 1, Process 1 detail diagram process box 2
   1.2. Level 1, Process box 2
      1.2.1. Level 1, Process 2 detail diagram process box 1
      1.2.2. Level 1, Process 2 detail diagram process box 2
   1.3. Level 1, Process box 3

Verifying and Validating Process Maps

Verification is the act of confirming accuracy or truth of some item. *Validation* is the act gaining confirmation or official sanction.[1] In terms of process improvement projects, verification is done with individual stakeholders, usually those who were interviewed to provide the information that formed the basis for the process diagrams. Validation is done via a *walk-through*, which is a structured meeting to review the verified process maps and officially sanction them.

Process Map Verification

The purpose of process map verification is to ensure that the interviewees and people who perform tasks in a process agree about the details and substance of the process. Apply the principal of triangulation to the review of processes (i.e., three people should verify every process step, conclusions about summary processes, and the placement of process steps in swimlanes).

Interviewees should have an opportunity to review the process maps developed as an outcome of their consultation. Within 2 days of an interview, the interviewee should be given a copy of the interview notes and any diagrams developed from the discussion. Not all discussions lead directly to process maps, but when enough information has been collected to develop process maps, the reviews should be completed by each interviewee as soon as possible.

In addition, several—usually 3 to 7—people who actually perform the work should review process maps to ensure that correct and complete information was given. This can be via e-mail and remote review as with interviewees or in person in a single group meeting or in individual meetings. Each of these modes of review is useful, but they are likely to result in different results unless everyone involved is highly motivated to participate in the project. Without having had interviews, e-mail and remote review are unlikely to result in more than superficial comments and are less likely to result in problem finding. Similarly, individual one-on-one reviews are likely to result in the deepest level of comments, but only one person's verification is acquired. Group meetings can result in the best outcome but may not be warranted as they may also take more time for users than individual meetings. For all types of sessions, project background, goals, and discussion of results to date (that are not confidential) should be a prelude to any feedback requests. These sessions serve to find errors and correct them before the validation meeting. The other alternative is to both verify and validate at the same time.

Process Map Validation Using Walk-Throughs

After process maps are complete and verified, a walk-through should be held to validate the maps. The steps involved in a walk-through are as follows:

1. Premeeting
   a. Gather the process maps and any process descriptions developed to explain the diagrams.
   b. Develop an agenda for the meeting.
   c. Invite all participants, sending them the agenda, process map(s), and other material to be reviewed.
2. Walk-through meeting
   a. A senior analyst conducts the walk-through by reviewing each diagram in detail.
   b. A project team member acts as scribe, noting any errors found. No attempt is made to fix errors.

c. A third person acts as timekeeper to ensure that the meeting meets its goals and stops at a designated time, usually about 1 hour.
d. Other participants include the process experts, managers interviewed, workers used to verify the process maps, or other stakeholders in process accuracy (e.g., internal auditors).
e. Meeting participants find errors of logic, syntax, or semantics in the diagram(s) being reviewed.
f. At the end of the meeting, the group decides if another group meeting is needed (i.e., the errors are severe or numerous) or if the corrections can be reviewed via e-mail.

3. Postmeeting
   a. The scribe types notes and circulates them to all participants.
   b. The project team members work through the errors found, addressing and rectifying each.
   c. As errors are corrected, a notation is made on the error list as to where the correction is placed within the diagram set.
   d. When all errors have been addressed, the outcomes are documented and circulated with corrected process maps to all participants to ensure correct and complete updating.

The walk-through can serve two purposes: one, to verify the process maps in a group setting, and two, to validate the maps, thus defining the "official" version of the processes.

Common Errors and Corrections

This section presents correct techniques to help avoid common novice errors. The most common errors include the following:

- incorrect role identification
- incorrect arrow placement
- incomplete handoffs
- incorrect handoffs to applications or databases
- dangling processes, connectors, or decision outcomes
- misnumbered connectors
- sloppy diagram style

Incorrect Role Identification

Misnamed or incorrectly named roles are a common problem. When a role is wrong, all icons in the swimlane are also likely to be wrong. Therefore, good practice is to ensure that roles are properly named before beginning the process map. Roles can be individual names, a job role, an organization, or an application—some entity that can perform independent actions. A database does not follow that definition but is a special case of a role when application support for a process is being analyzed.

Disk drives that contain shared documents are not swimlanes because a disk drive cannot, without direction from an application, perform independent actions. The shared repository could be a role, but care would be needed to ensure that the commands for actions were clearly differentiated as human or application process steps.

Another error concerning the definition of roles is use of an individual's name (e.g., "Sharon"). Sharon may be the person who currently holds responsibility for some tasks, but she would be representative of a title or role in that regard. The role name is preferred over a person's name. The same reasoning holds if Sharon is a supervisor who has added duties beyond those normally performed. The title "supervisor" would be used to distinguish the role.

Incorrect Arrow Placement

There are two ways that arrows are used incorrectly—the first is to have them crossing each other; the second is to violate the rules for numbers of arrows in and out of each icon type.

The "before" side of Figure 4.22 depicts crossing arrows and, because of the arrows crossing, the author was unaware that the process was dangling (i.e., it did not have any out arrow). The "after" side of Figure 4.22 shows the fixed diagram with no crossing arrows and the dangling process now connecting to the "stop" icon.

The second type of arrow problem violates the rules of arrows in or out for the icon type. Several wrong examples are shown in Figure 4.23. In the "before" side of the map, the two arrows out of the "save schedule" process and the two arrows out of the next process violate the rule of one arrow out of a process box. To fix the problem, diverging arrows emanate

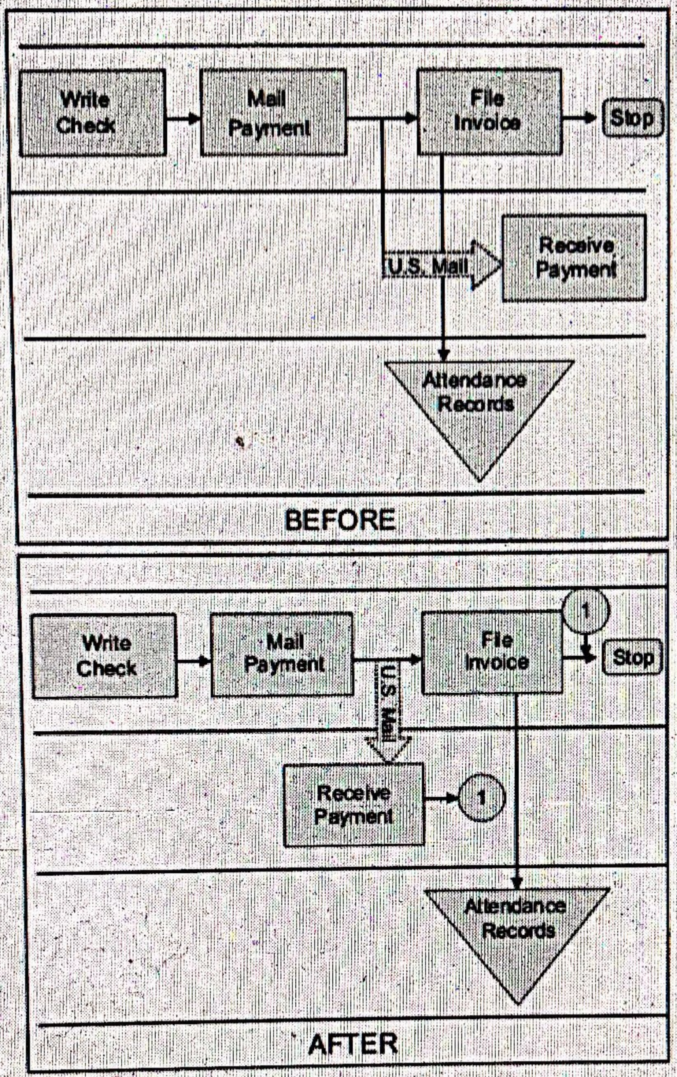


Figure 4.22. Incorrect arrow placement.

from one of the "out" arrows. The "after" side of the figure shows how to fix these problems. There is another unrelated problem with this diagram. Can you find it? The answer is at the end of the chapter.[2]

Incomplete Handoffs

Handoffs seem to cause significant problems for novice process mappers. If the sending side of a handoff shows the preparation of an item, then the receiving side must show receipt of the item. Handoffs are important because they are frequently the source of problems such as lost material

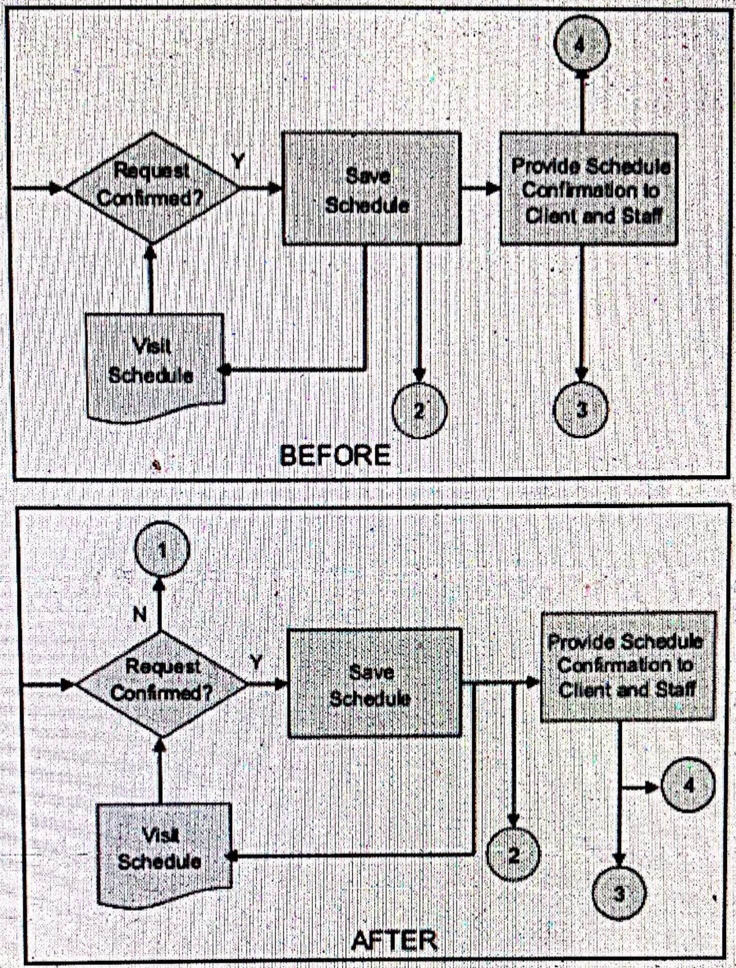


Figure 4.23. Example of wrong arrow use.

or documents, delivery delays, or technology bugs. Handoffs, especially those across organizational boundaries, represent the "white space" on an organization chart (or process map) and are error sources because no one has responsibility for monitoring them.

One handoff problem is deciding whether or not the handoff is a joint activity, requiring the outer multilane process box. Think of the top swimlane in Figure 4.24 as referring to a "client" and the bottom swimlane referring to a "vendor." The "before" diagram shows a handoff of payroll checks as being a joint activity. This would be correct if the delivery is in person and both sides of the handoff in fact participate in the delivery and receipt. If not (e.g., if the checks were delivered electronically or via mail), then the "after" diagram would be correct. Thus for this example, both before and after diagrams *could* be correct, depending on the context.

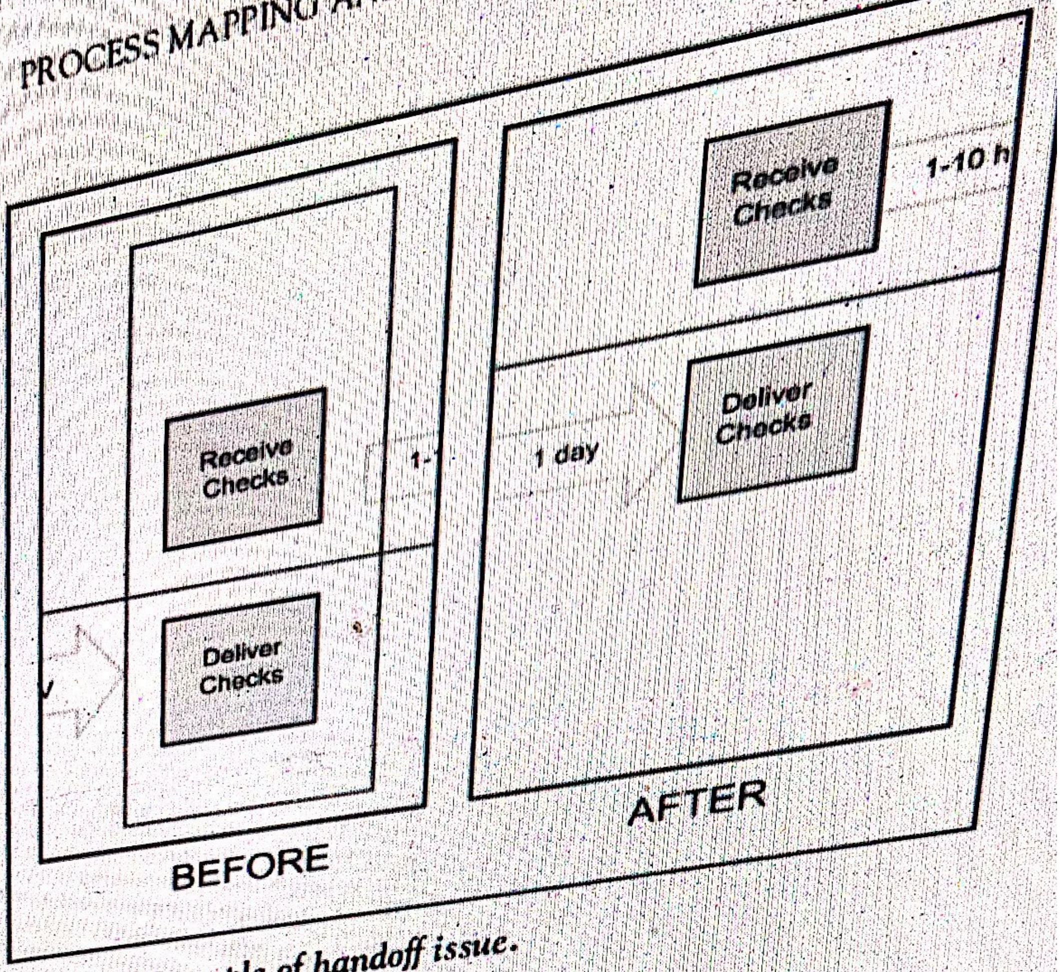


Figure 4.24. Example of handoff issue.

A different type of problem is shown in Figure 4.25. In the "before" map, the author assumed that passing inspection was acceptable as the start of a handoff. This is wrong. When inspection is passed, the inspector gives approval to the person receiving it, as shown in the "after" side of the figure. The left side is wrong because giving specific approval implies that some verbal approval, e-mail, piece of paper, or other evidence of approval is provided. The left side does not provide for the passing of anything to the receiver. Therefore, nothing can be received. To correct this situation, the passing of inspection occurs, and then the "give inspection approval" task occurs in order to start the handoff. The receiving side of the handoff is correct in both diagrams.

Incorrect Handoffs to Applications or Databases

Handoffs to applications and databases are a bit trickier because not only one send-receive interaction takes place but also a second send-receive is required to display results and the next steps of a process. In the middle of the two handoffs, the processing takes place. The processing, if simple,

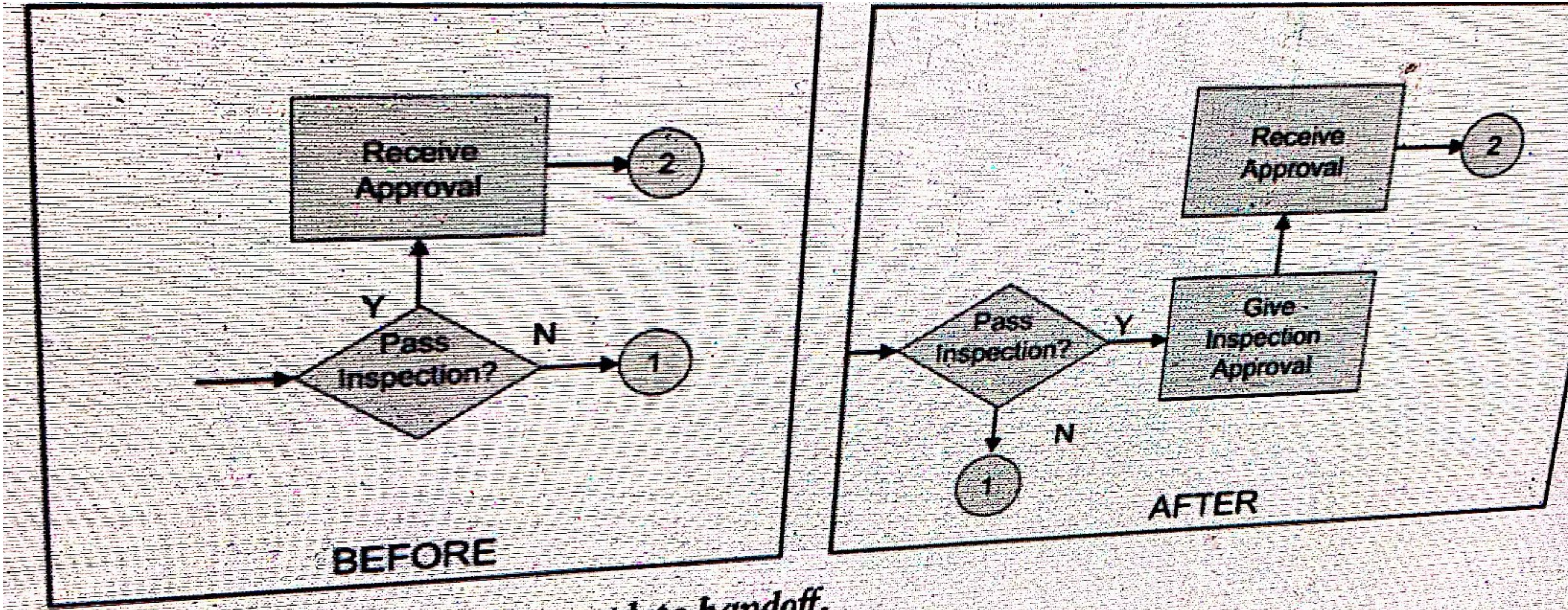


Figure 4.25. Example of incomplete handoff.

can be depicted within a database icon. If more complex or application work is completed before the database is involved (e.g., computations on data are performed), then more process steps are shown. Figure 4.26 shows both application and database handoff errors.

The first error in Figure 4.26 relates to "retrieves entry ticket." The actual handoff, "get-retrieve," is acceptable, although the wording could be improved by changing the warehouse manager's entry to "request entry ticket" and the database system entry to "retrieve (or get) ET request." This revised wording more clearly identifies a handoff. The problem is that the "retrieves" process output should be the step in which the data are obtained, followed by a process step to display the response, which is then followed by a process step to get the response, leading to the creation of a new pickup note entry.

The second error, "saves new pickup notes entry," occurs at the database end of the process. This entry should be a receipt of the pickup note request, followed by the process of creating a new entry with a database icon, followed by a display of the new pickup note status. An improvement from "database" as a swimlane name would be to identify the application as the "pickup note application." The corrected diagram is shown in Figure 4.27

While this diagram is quite a bit more complicated, it is also much clearer. Every process box in the "entry ticket" and "pickup note" application swimlanes requires some programming to accomplish those tasks. If the handoff processing is omitted or abbreviated on the process map, then it is likely that programming errors would occur. In addition, omitting these handoffs on the current process map would result in incomplete analysis and potentially missed opportunities for improvement.

Dangling Processes, Connectors, or Decision Outcomes

Often, it is easy to forget to complete temporarily incomplete areas of a diagram as the main logic is being developed. Figure 4.28 shows two common errors. This summary diagram has a connector that is unused in the rest of the diagram, and the "daily process" is unconnected to anything else. Another common error not in this example would be to omit "yes" and "no" outcomes for a decision. After drawing a diagram, as part of its team review, one person should be tasked with ensuring that all connections and lines are correct and that the diagram is syntactically

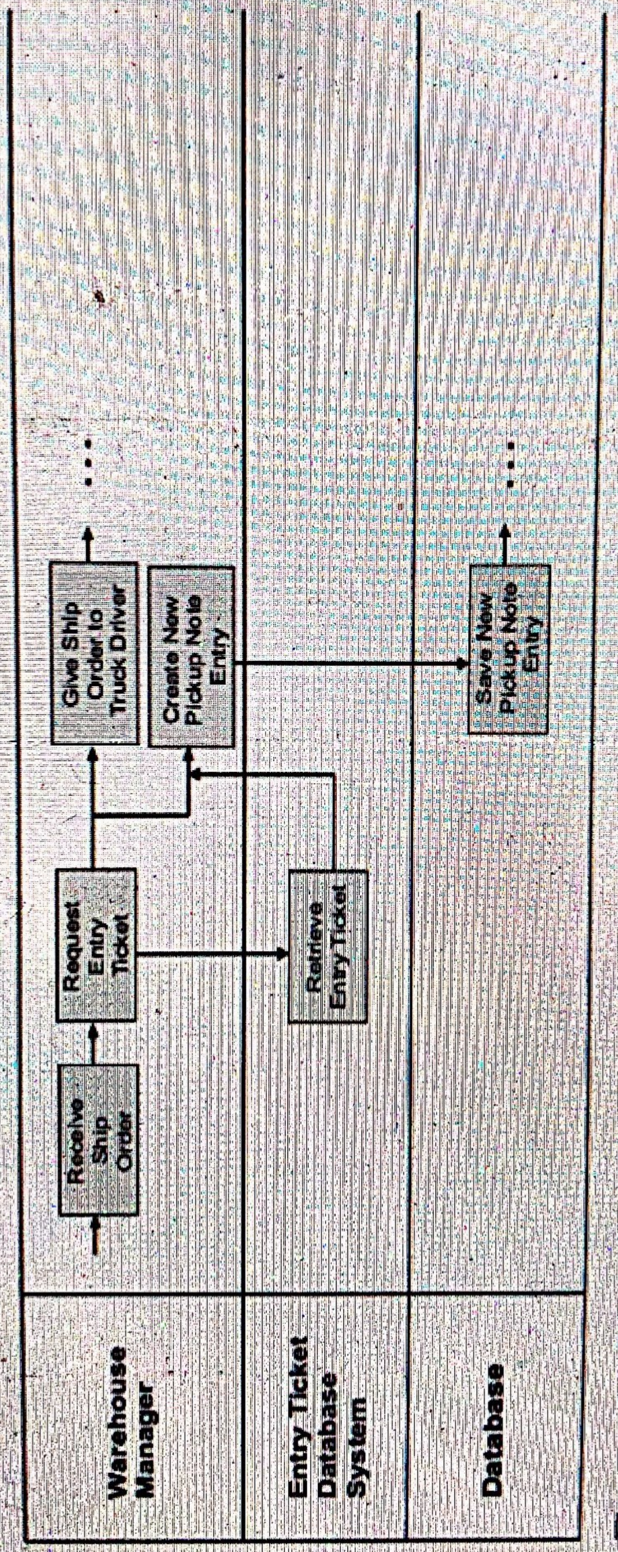
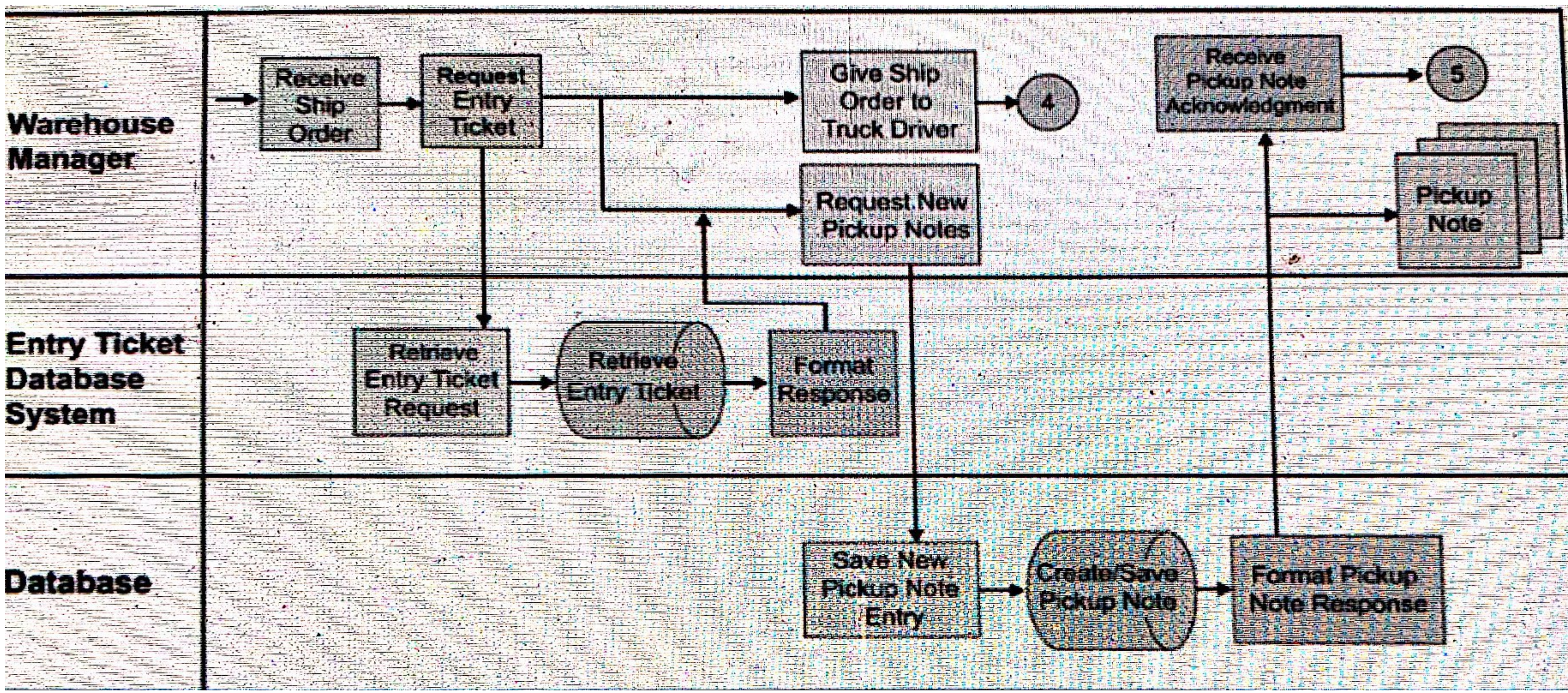


Figure 4.26. Incorrect application and DB handoffs.

Scanned with CamScanner



Scanned with CamScanner

accurate. If the syntax (i.e., the icons and line connections) are not correct, then by definition, the drawing is wrong.

Figure 4.29 shows a corrected diagram for these errors. Of course, it seems obvious on a summary diagram such as this what the errors are, but on diagrams that can run 30 pages or more, the errors become insidious and difficult to track.

Misnumbered Connectors

Connectors come in pairs. If there is a "1" connector out of a process, there must then be a "1" connector into some other decision or process. Figures 4.28 and 4.29 show common connector errors. In Figure 4.28, the "1" connector is unmatched. This is the most common error—dangling connectors. This type of error often occurs when deferring some logic while developing the main logic path of the map. It is easily fixed during a group review when someone is given responsibility for monitoring that all connectors come in pairs.

The second common error is the use of sequential numbers—for example, assuming that the "1" connector out of a process is paired with a sequentially numbered "2" connector into a process or decision. This mistake in logic is usually made by novices who have not drawn many diagrams. It is easily corrected by remembering that every number used *out* of a process must be paired with the same number *into* a process or decision.

Sloppy Diagram Style

Novices tend to believe that any syntactically correct diagram is an acceptable one. However, diagrams that use different-sized icons, different length arrows, sloppy telecom arrows, unexplained lines, and incorrect icons are difficult to read. In addition, diagrams that do not separate swimlanes so that they are easily readable and those that use different-sized swimlanes not only look unkempt but also are also difficult to read.

In Figure 4.30, the top ("before") map has all of the errors mentioned previously and takes three pages to print legibly. The bottom ("after") map has the errors fixed and takes only two pages to print.

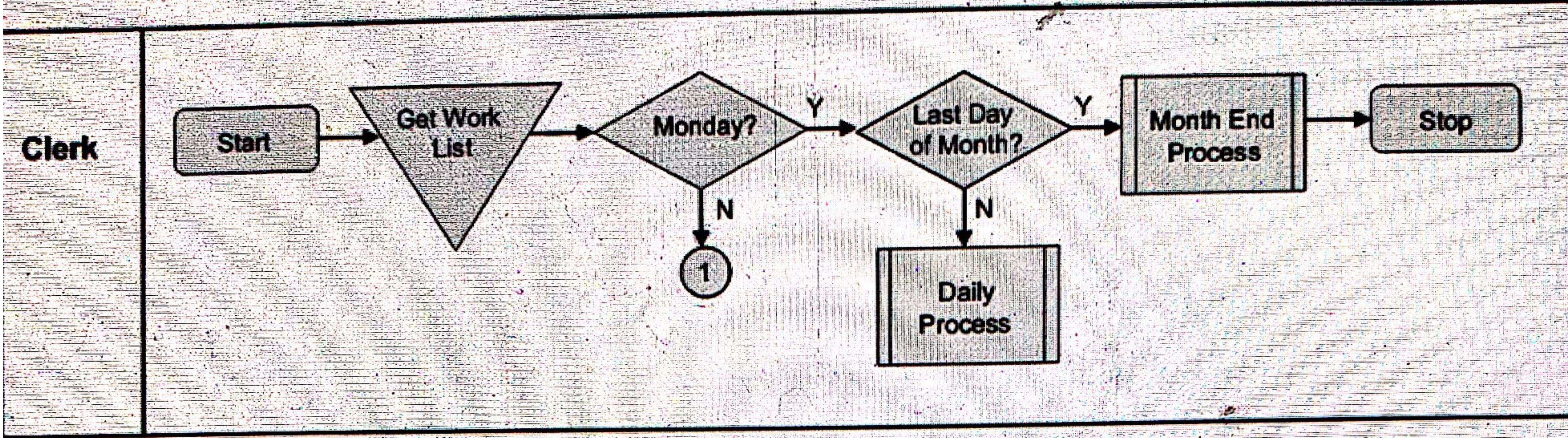
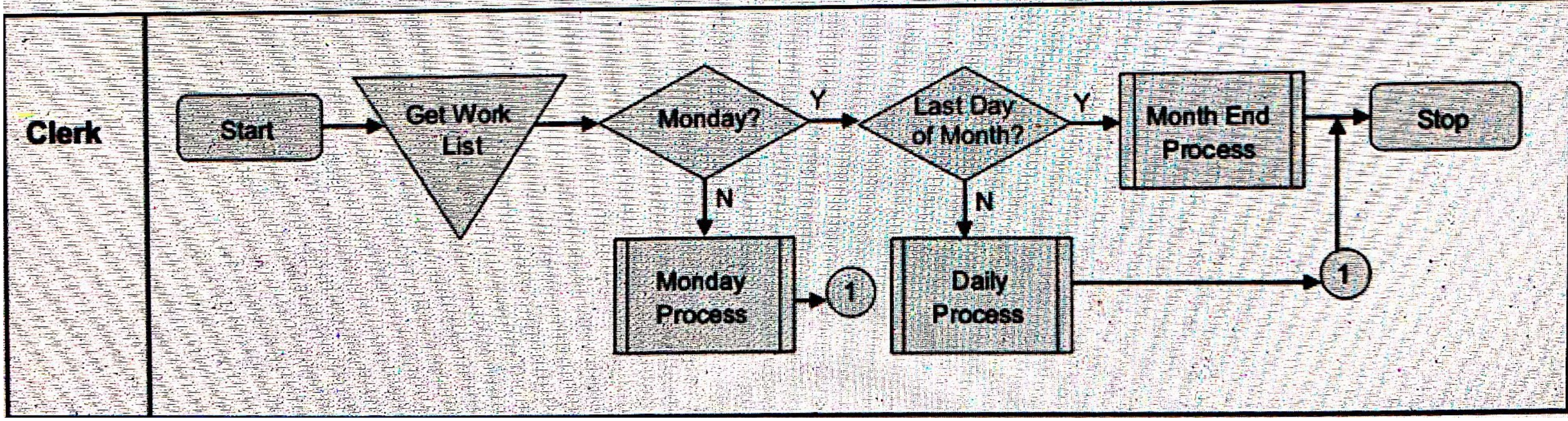


Figure 4.28. Dangling icons.



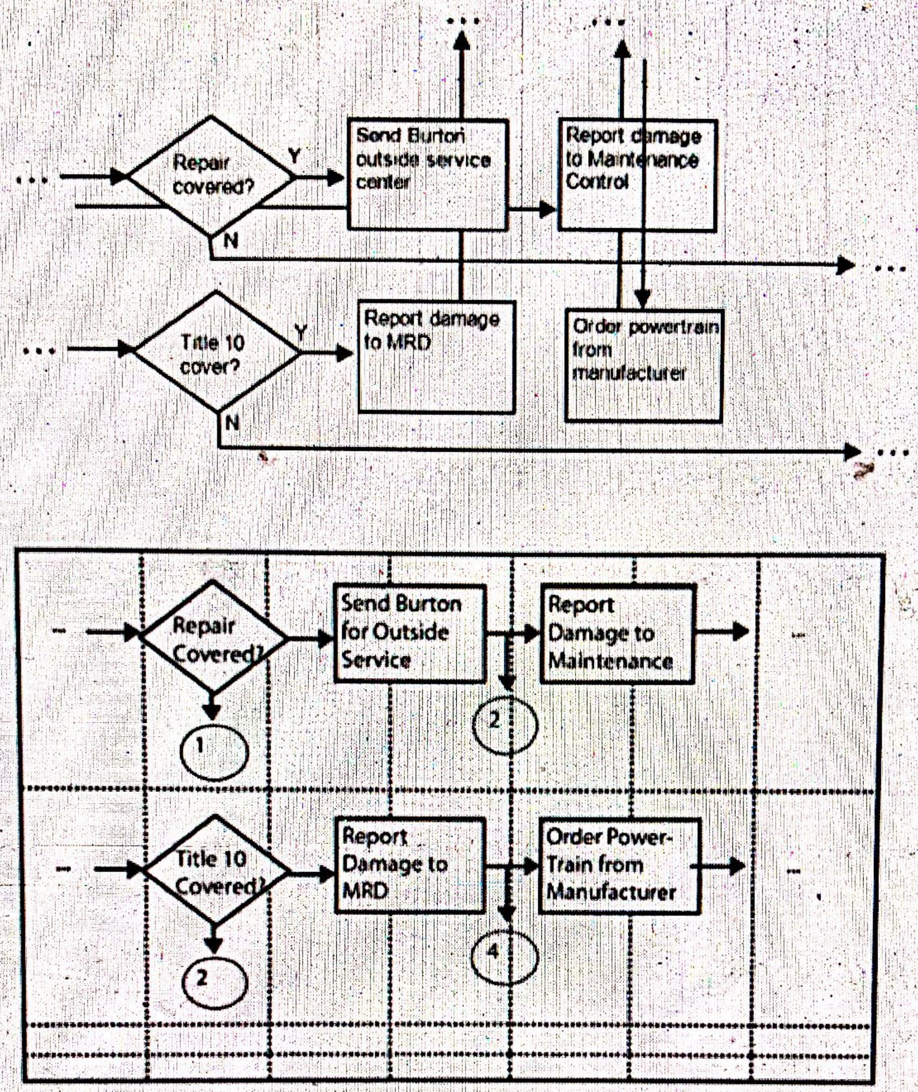


Figure 4.30. Before (top) and after (bottom) of fixes for sloppy mapping style.

Summary

Multiple inputs, outputs, or methods for conducting a process should all be clearly shown on the process map. In the case of multiple methods for conducting a process, either a decision diamond is shown to lead to the two paths, or, if the process differences relate to individuals—for example, a clerk and a manager—the swimlane is divided into sublanes to indicate the different flows.

A small, circular connector is used to show a change of process flow and to connect to the point where the flow continues. Connectors increase diagram clarity in situations where there are many possible paths through a process, minimize stop points for a process, or, in complex diagrams, minimize crossing lines.

Time, measurement points, goals, or other significant items can be shown by vertical lines through a map, when relevant.

Verifying and validating a process diagram are both important. First, diagrams should be reviewed with each interviewee and the people who do the work to confirm interview results. Inconsistencies or disagreements should be corrected or the alternatives shown on the diagram. Similarly, suspected errors should be highlighted on the diagram. Group walk-throughs complete the validation of diagrams. A walk-through is a formal group meeting with the goal of finding errors. A team member (not the author) reviews material with stakeholders, and project team members fill scribe and timekeeper roles. The stakeholders are the primary finders of errors.

In many ways, process mapping is more of an "art" than a "science." However, once the details of a process are known, maps by different people should be essentially the same, differing only in the level of detail shown on a diagram. When drawing maps, do not become a victim of the rules; for instance, do not expect to know all of the details. Use process map variations such as those in this chapter to explain processes more clearly. Process maps should make the process easier to understand rather than more difficult. Novice errors to avoid include incorrect role identification, arrow placement, handoffs, dangling icons, and sloppy style.

PART II

The Middle Game

The middle game in chess relies on the skill of the player to know the sequences of moves and combinations of approaches most likely to be effective. Similarly, a process improvement project team is required to have many skills and techniques available from which they choose those more likely to arrive at a successful project conclusion in the shortest amount of time. The middle game is "where the magic happens"—where expertise and know-how determine the quality of assessment, redesign, and project outcome. This part of the book begins the discussion of methods for analysis and change. In chapter 5, brainstorming techniques that are useful in many situations are discussed, along with techniques used to identify processes for change. In chapter 6, we discuss techniques to aid in removing non-value-adding process steps and other techniques for "leaning" the process. Chapter 7 is a review of analysis techniques for "cleaning" the remaining process steps to ensure that they also are lean and contain no unneeded actions. Chapter 8 addresses "greening" of the process to improve its environmental friendliness for your organization and to remove as much of the process from human hands as feasible. At the end of this part, the process and its problems will have been evaluated from many different perspectives, each of which adds value to the final recommendations.

Keep in mind that at the end of this part of the project, the outcome is not yet clear and that there may be conflicting or irreconcilable recommendations that come from the various analyses conducted. These issues are resolved in part III of the book.