Chapter 6
Qualitative Process Analysis

Quality is free, but only to those who are willing to pay heavily for it.
Tom DeMarco (1940–)

Analyzing business processes is both an art and a science. In this respect, qualitative analysis is the artistic side of process analysis. Like fine arts, such as painting, there is not a single way of producing a good process analysis, but rather a range of principles and techniques that tell us what practices typically lead to a "good" process analysis. When learning to paint, you learn how to hold the brush, how to produce different types of brushstroke, how to mix colors, etc. The rest of the art of painting is up to you to acquire by means of practice, discernment and critical self-assessment.

In this chapter, we introduce a few basic principles and techniques for qualitative process analysis. First, we present principles aimed at making the process leaner by identifying unnecessary parts of the process in view of their elimination. Next, we present techniques to identify and analyze the weak parts of the process, meaning the parts that create issues that negatively affect the performance of the process. In particular, we discuss how to analyze the impact of issues in order to prioritize redesign efforts.

6.1 Value-Added Analysis

Value-added analysis typically consists of two stages: value classification and waste elimination. Below we discuss each of these stages in turn.

6.1.1 Value Classification

Value-added analysis is a technique aimed at identifying unnecessary steps in a process in view of eliminating them. In this context, a *step* may be a task in the process, or part of a task, or a handover between two tasks. For example, if a task "Check

M. Dumas et al., *Fundamentals of Business Process Management*,
DOI 10.1007/978-3-642-33143-5_6,

purchase order" ends with the Purchase Order (PO) being sent by internal mail to a supervisor, and the next task "Approve purchase order" starts when the supervisor receives and checks the PO, then we can say that the transportation of the PO via internal mail is a step—a potentially unnecessary (non-value-adding) step in this context. It is often the case that one task involves several steps. For example, a task "Check invoice" may involve the following steps:

1. Retrieve the PO(s) that corresponds to the invoice.
2. Check that the amounts in the invoice and those in the PO coincide.
3. Check that the products or services referenced in the PO have been delivered.
4. Check that the supplier's name and banking details in the invoice coincide with those recorded in the Supplier Management System.

In some cases, steps within a task are documented in the form of checklists. The checklists tell the process participants what things need to be in place before a task is considered to be complete. If detailed checklists are available, the process analyst can use them to decompose tasks into steps. Unfortunately, such checklists are not always available. In many cases, process participants have an implicit understanding of the steps in a task because they perform the task day in and day out. But this implicit understanding is not documented anywhere. In the absence of such documentation, the process analyst needs to decompose each task into steps by means of observation and interviewing.

Having decomposed the process into steps, a second prerequisite for value-added analysis is to identify who is the customer of the process and what are the positive outcomes that the customer seeks from the process (cf. Sect. 1.2). These outcomes are said to add value to the customer, in the sense that fulfilling these outcomes is in the interest or for the benefit of the customers.

Having decomposed the process into steps and having clearly identified the positive outcomes of a process, we can then analyze each step in terms of the value it adds. Steps that directly contribute to positive outcomes are called *value-adding steps*. For example, consider a process for repairing a washing machine or other domestic appliance. The steps in this process where the technician diagnoses the problem with the machine are clearly value-adding, as it directly contributes to the outcome the customer wishes to see, that is, that the machine is repaired. Also, the steps related to repairing the machine are value-adding.

Some steps do not directly add value to the customer but they are necessary for the business. Consider again the example of a process for repairing a washing machine. Imagine that this process includes a step "Record defect" in which the technician enters data about the washing machine and an explanation of the defect found in a washing machine into an information system. This step per se is not value-adding for the customer. The customer wishes the machine to be fixed and does not get value by the fact that the defect in their machine was recorded in an information system of the repairing company. However, recording defects and their resolution helps the company to build up a knowledge base of typical defects and their resolution. This knowledge base is extremely valuable when new technicians are recruited in the company, since they can learn from knowledge that more experienced technicians have recorded. Also, such information allows the company to

detect frequent defects and to report such defects to the manufacturer or distributor of the washing machine. Steps such as "Record defect" are termed *business value-adding steps*, in the sense that the customer is not willing to pay for the performance of these steps nor do they gain satisfaction from these steps being performed (so the steps are not value-adding) but the step is necessary or useful to the company that performs the process.

In summary, value-added analysis is a technique whereby an analyst decorticates a process model, extracts every step in the process and classifies these steps into one of three categories, namely:

- Value-adding (VA): This is a step that produces value or satisfaction vis-à-vis of the customer. When determining whether or not a step is value-adding, it may help to ask the following question: Would the customer be willing to pay for this activity?
- Business value-adding (BVA): The step is necessary or useful for the business to run smoothly, or it is required due to the regulatory environment of the business.
- Non-value adding (NVA): The step does not fall into any of the other two categories.

Example 6.1 We consider the process for equipment rental described in Example 1.1 (p. 2). As discussed in Sect. 1.2, the customer of this process is the site engineer who submits an equipment rental request. From the perspective of the site engineer, the positive outcome of the process is that the required piece of equipment is available in the construction site when needed. Let us analyze the fragment of this process described in Fig. 1.6. To identify the relevant steps, we will decorticate the model task by task. While we do this, we will also classify the steps into VA, BVA and NVA.

- The first task in the process model is the one where the engineer lodges the request. From the description in Example 1.1, we observe there are three steps in this task:

  1. Site engineer fills in the request.
  2. Site engineer sends the request to the clerk via e-mail (handover step).
  3. Clerk opens and reads the request (handover step).

  Arguably, filling the request is value-adding insofar as the site engineer cannot expect the equipment to be rented if they do not ask for it. In one way or another, the site engineer has to request the equipment in order to obtain it. On the other hand, the site engineer does not get value out of sending the request to the clerk by e-mail nor do they get value out of the clerk having to open and read the request. More generally, handover steps between process participants, such as sending and receiving internal messages, are not value-adding.
- The second task is the one where the clerk selects a suitable equipment from the supplier's catalog. We can treat this task as a single step. This step is value-adding insofar as it contributes to identifying a suitable equipment to fulfill the needs of the site engineer.

- In the third task, the clerk calls the supplier to check the availability of the selected equipment. Again, we can treat this task as a single step. This step is value-adding insofar as it contributes to identifying a suitable and available equipment. If the equipment is available, the clerk will recommend that this equipment be rented. To this end, the clerk adds the details of the recommended equipment and supplier to the rental request form and forwards the form to the works engineer for approval. Thus we have two more steps: (i) adding the details to the rental request and (ii) forwarding the rental request to the works engineer. The first of these steps is business value-adding since it helps the company to keep track of the equipment they rent and the suppliers they rent from. Maintaining this information is valuable when it comes to negotiating or re-negotiating bulk agreements with suppliers. The handover between the clerk and the works engineer is not value-adding.
- Next, the works engineer examines the rental request in view of approving it or rejecting it. We can treat this examination as one step. This step is a *control step*, that is, a step where a process participant or a software application checks that something has been done correctly. In this case, this control step helps the company to ensure that equipment is only rented when it is needed and that the expenditure for equipment rental in a given construction project stays within the project's budget. Control steps are generally business value-adding, although an analyst may ask the question of how many control steps are needed and how often they should be performed.
- If the works engineer has an issue with the rental request, the works engineer communicates it to the clerk or the site engineer. This communication is another step and it is business value-adding since it contributes to identifying and avoiding misunderstandings within the company. If approved, the request is sent back to the clerk; this is a handover step and it is thus non-value-adding.
- Finally, assuming the request is approved, the clerk produces and sends the PO. Here we can identify two more steps: produce the PO and send the PO to the corresponding supplier. Producing the PO is business value-adding. It is necessary in order to ensure that the rental request cost is correctly accounted for and eventually paid for. Sending the PO is value-adding: It is this act that makes the supplier know when the equipment has to be delivered on a given date. If the supplier did not get this information, the equipment would not be delivered. Note, however, that what is value-adding is the fact that the supplier is explicitly requested by the construction company to deliver the equipment on a given date. The fact that this request is made by sending a PO is secondary in terms of adding value to the site engineer.

The identified steps and their classification are summarized in Table 6.1.

Classifying steps into VA, BVA and NVA is to some extent subjective and depends on the context. For example, one may question whether producing the PO is a VA or a BVA step. Arguably, in order for the equipment to be available, the supplier needs to have an assurance that the equipment rental fee will be paid. So one could say that the production of the PO contributes to the rental of the equipment

Table 6.1 Classification of steps in the equipment rental process

Step	Performer	Classification
Fill request	Site engineer	VA
Send request to clerk	Site engineer	NVA
Open and read request	Clerk	NVA
Select suitable equipment	Clerk	VA
Check equipment availability	Clerk	VA
Record recommended equipment & supplier	Clerk	VA
Forward request to works engineer	Clerk	VA
Open and examine request	Works engineer	BVA
Communicate issues	Works engineer	BVA
Forward request back to clerk	Works engineer	NVA
Produce PO	Clerk	BVA
Send PO to supplier	Clerk	BVA

since the PO serves to assure the supplier that the payment for the rental equipment will be made. However, as mentioned above, what adds value to the site engineer is the fact that the supplier is notified that the equipment should be delivered at the required date. Whether this notification is done by means of a PO or by means of a simple electronic message sent to the supplier is irrelevant, so long as the equipment is delivered. Thus, producing a formal document (a formal PO) is arguably not value-adding. It is rather a mechanism to ensure that the construction company's financial processes run smoothly and to avoid disputes with suppliers, e.g. avoiding the situation where a supplier delivers a piece of equipment that is not needed and then asks for payment of the rental fee. More generally, we will take the convention that steps imposed by accounting or legal requirements are BVA, even though one could argue differently in some cases.

Exercise 6.1 Consider the process for university admission described in Exercise 1.1 (p. 4). What steps can you extract from this process? Classify these steps into VA, BVA and NVA.

6.1.2 Waste Elimination

Having identified and classified the steps of the process as discussed above, one can then proceed to determining how to eliminate waste. A general rule is that one should strive to minimize or eliminate NVA steps. Some NVA steps can be eliminated by means of automation. This is the case of handovers for example, which can be eliminated by putting in place an information system that allows all stakeholders to know what they need to do in order to move forward the rental requests. When

the site engineer submits a rental request via this information system, the request would automatically appear in the to-do list of the clerk. Similarly, when the clerk records the recommended supplier and equipment, the works engineer would be notified and directed to the request. This form of automation makes these NVA steps transparent to the performers of the steps. The topic of process automation will be discussed in further detail in Chap. 9.

A more radical approach to eliminating NVA steps in the working example is to eliminate the clerk altogether from the process. This means moving some of the work to the site engineer so that there are less handovers in the process. Of course, the consequences of this change in terms of added workload to the site engineer need to be carefully considered. Yet another approach to eliminate NVA (and BVA) steps would be to eliminate the need for approval of rental requests in cases where the estimated cost is below a certain threshold. Again, this option should be weighted against the possible consequences of having less control steps in place. In particular, if the site engineers were given full discretion to rent equipment at their own will, there would need to be a mechanism in place to make them accountable in case they rent unnecessary equipment or they rent equipment for excessively and unnecessarily long periods. Such process redesign questions will be further discussed in Chap. 8.

While elimination of NVA steps is generally considered a desirable goal, elimination of BVA steps should be considered as a trade-off given that BVA steps play a role in the business. Prior to eliminating BVA steps, one should first map BVA steps to business goals and business requirements, such as regulations that the company must comply to and risks that the company seeks to minimize. Given a mapping between BVA steps on the one hand and business goals and requirements on the other, the question then becomes the following: What is the minimum amount of work required in order to perform the process to the satisfaction of the customer, while fulfilling the goals and requirements associated to the BVA steps in the process? The answer to this question is a starting point for process redesign.

6.2 Root Cause Analysis

When analyzing a business process, it is worth keeping in mind that "even a good process can be made better" [28]. Experience shows that any non-trivial business process, no matter how much improvement it has undergone, suffers from a number of issues. There are always errors, misunderstandings, incidents, unnecessary steps and other forms of waste when a business process is performed on a day-to-day basis.

Part of the job of a process analyst is to identify and to document the *issues* that plague a process. To this end, an analyst will typically gather data from multiple sources and will interview several stakeholders, chiefly the process participants but also the process owner and managers of organizational units involved in the process. Each stakeholder has a different view on the process and will naturally have a tendency to raise issues from their own perspective. The same issue may be perceived

differently by two stakeholders. For example, an executive manager or a process owner will typically see issues in terms of performance objectives not being met or in terms of constraints imposed for example by external pressures (e.g. regulatory or compliance issues). Meanwhile, process participants might complain about insufficient resources, hectic timelines as well as errors or exceptions perceived to be caused by other process participants or by customers.

Root cause analysis is a family of techniques to help analysts identify and understand the root cause(s) of problems or undesirable events. Root cause analysis is not confined to business process analysis. In fact, root cause analysis is commonly used in the context of accident or incident analysis as well as in manufacturing processes where it is used to understand the root cause of defects in a product. In the context of business process analysis, root cause analysis is helpful to identify and to understand the issues that prevent a process from having a better performance.

Root cause analysis encompasses a variety of techniques. In general, these methods include guidelines for interviewing and conducting workshops with relevant stakeholders, as well as techniques to organize and to document the ideas generated during these interviews or workshops. Below, we will discuss two of these techniques, namely *cause-and-effect diagrams* and *why–why diagrams*.

6.2.1 Cause–Effect Diagrams

Cause–effect diagrams depict the relationship between a given *negative effect* and its causes. In the context of process analysis, a negative effect is usually either a recurrent issue or an undesirable level of process performance. Causes can be divided into causal and contributing factors (hereby called *factors*) as explained in the box below.

CAUSAL VERSUS CONTRIBUTING FACTORS

Two broad types of cause are generally distinguished in the area of root cause analysis, namely *causal factors* and *contributing factors*. Causal factors are those factors that, if corrected, eliminated or avoided would prevent the issue from occurring in future. For example, in the context of an insurance claims handling process, errors in the estimation of damages lead to incorrect claim assessments. If the damage estimation errors were eliminated, a number of occurrences of the issue "Incorrect claim assessment" would definitely be prevented. Contributing factors are those that set the stage for, or that increase the chances of a given issue occurring. For example, consider the case where the user interface for lodging the insurance claims requires the claimant to enter a few dates (e.g. the date when the claim incident occurred), but the interface does not provide a calendar widget so that the user can easily select the date. This deficiency in the user interface may increase the chances that

the user enters the wrong date. In other words, this deficiency contributes to the issue "Incorrect claim data entry".

While the distinction between causal and contributing factor is generally useful when investigating specific incidents (for example investigating the causes of a given road accident), the distinction is often not relevant or not sufficiently sharp in the context of business process analysis. Accordingly, in this chapter we will use the term *factor* to refer to causal and contributing factors collectively.

In a cause–effect diagram, factors are grouped into categories and possibly also sub-categories. These categories are useful in order to guide the search for causes. For example, when organizing a brainstorming session for root cause analysis, one way to structure the session is to first go around the table asking each participant to give their opinion on possible causes of the issue at hand. The causes are first written down in any order. Next, the identified causes are classified according to certain categories and the discussion continues in a more structured way using these categories as a framework.

A well-known categorization for cause–effect analysis are the so-called 6M's, which are described below together with possible sub-categorizations.

1. **Machine** (technology)—factors pertaining to the technology used, like for example software failures, network failures or system crashes that may occur in the information systems that support a business process. A useful sub-categorization of Machine factors is the following:
   a. Lack of functionality in application systems.
   b. Redundant storage of data across systems, leading for example to double data entry (same data entered twice in different systems) and data inconsistencies across systems.
   c. Low performance of IT of network systems, leading for example to low response times for customers and process participants.
   d. Poor user interface design, leading for example to customers or process participants not realizing that some data are missing or that some data are provided but not easily visible.
   e. Lack of integration between multiple systems within the enterprise or with external systems such as a supplier's information system or a customer's information system.
2. **Method** (process)—factors stemming from the way the process is defined or understood or in the way it is performed. An example of this is when a given process participant A thinks that another participant B will send an e-mail to a customer, but participant B does not send it because they are not aware they have to send it. Possible sub-categories of Method factors include:
   a. Unclear, unsuitable or inconsistent assignment of decision-making and processing responsibilities to process participants.

   b. Lack of empowerment of process participants, leading to process participants not being able to make necessary decisions without consulting several levels above in their organizational hierarchy. Conversely, excessive empowerment may lead to process participants having too much discretion and causing losses to the business through their actions.
   c. Lack of timely communication between process participants or between process participants and the customer.
3. **Material**—factors stemming from the raw materials, consumables or data required as input by the activities in the process, like for example incorrect data leading to a wrong decision being made during the execution of a process. The distinction between raw materials, consumables and data provides a possible sub-categorization of these factors.
4. **Man**—factors related to a wrong assessment or an incorrectly performed step, like for example a claims handler accepting a claim even though the data in the claim and the rules used for assessing the claim require that the claim be rejected. Possible sub-categories of Man factors include:
   a. Lack of training and clear instructions for process participants.
   b. Lack of incentive system to motivate process participants sufficiently.
   c. Expecting too much from process participants (e.g. overly hectic schedules).
   d. Inadequate recruitment of process participants.
5. **Measurement**—factors related to measurements or calculations made during the process. In the context of an insurance claim, an example of such a factor is one where the amount to be paid to the customer is miscalculated due to an inaccurate estimation of the damages being claimed.
6. **Milieu**—factors stemming from the environment in which the process is executed, like for example factors originating from the customer, suppliers or other external actors. Here, the originating actor is a possible sub-categorization. Generally, milieu factors are outside the control of the process participants, the process owner, and other company managers. For example, consider a process for handling insurance claims for car accidents. This process depends partly on data extracted from police reports (e.g. police reports produced when a major accident occurs). It may happen in this context that some errors during the claims handling process originate from inaccuracies or missing details in the police reports. These factors are to some extent outside the control of the insurance company. This example illustrates that milieu factors may need to be treated differently from other (internal) factors.

These categories are meant as guidelines for brainstorming during root cause analysis rather than gospel that should be followed to the letter. Other ways of categorizing factors may be equally useful. For example, one alternative categorization is known as the 4P's (Policies, Procedures, People, and Plant/Equipment). Also, it is sometimes useful to classify factors according to the activities in the process where they originate (i.e. one category per major activity in the process). This approach allows us to easily trace the relation between factors and activities in the process.

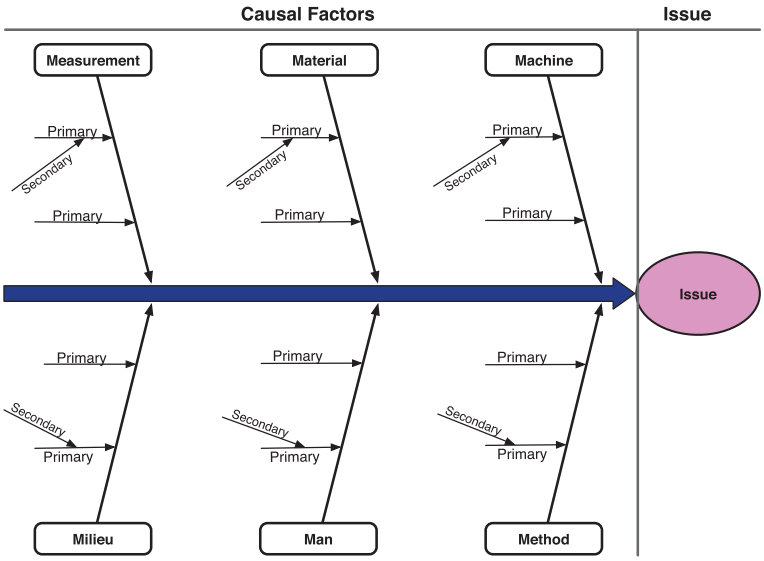


Fig. 6.1 Template of a cause–effect diagram based on the 6M's

The above categories are useful not only as a guide for brainstorming during root cause analysis, but also as a basis for documenting the root causes in the form of a cause–effect diagram. Concretely, a cause–effect diagram consists of a main horizontal line (the *trunk*) from which a number of branches stem (cf. Fig. 6.1). At one end of the trunk is a box containing the negative effect that is being analyzed (in our case the *issue* being analyzed). The trunk has a number of main branches corresponding to the categories of factors (e.g. the 6M's above). The root causes are written in the sub-branches. Sometimes, it is relevant to distinguish between *primary factors*, meaning factors that have a direct impact on the issue at hand, from *secondary factors*, which are factors that have an impact on the primary factors. For example, in the context of an insurance claims handling process, an inaccurate estimation of the damages leads to a miscalculation of the amount to be paid for a given claim. This inaccurate estimation of the damages may itself stem from a lack of incentive from the repairer to accurately calculate the cost of repairs. Thus, "Inaccurate damage estimation" can be seen as a primary factor for "Liability miscalculation", while "Lack of incentive to calculate repair costs accurately" is a secondary factor behind the "Inaccurate damage estimation". The distinction between primary and secondary factors is a first step towards identifying chains of factors behind an issue. We will see later in this chapter that why–why diagrams allow us to dig deeper into such chains of factors.

Because of their visual appearance, cause–effect diagrams are also known as *Fishbone diagrams*. Another common name for such diagrams is *Ishikawa diagrams* in allusion to one of its proponents—Kaoru Ishikawa—one of the pioneers of the field of quality management.

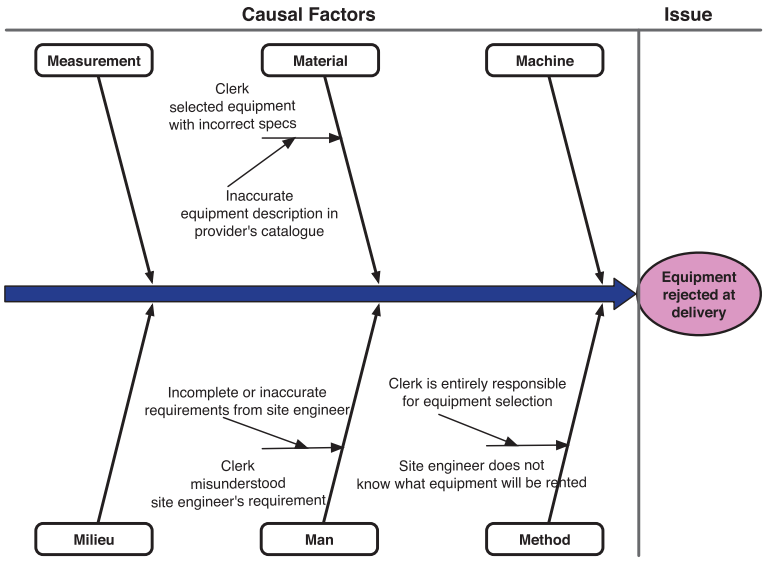


Fig. 6.2 Cause–effect diagram for issue "Equipment rejected at delivery"

Example 6.2 We consider again the equipment rental process described in Example 1.1 (p. 2). During an audit of this process, several issues were identified. It turns out that oftentimes the site engineer finds that the equipment delivered at the construction site is not suitable because it is either too small or not powerful enough for the job. Hence it has to be rejected. One clerk claims that the site engineers generally do not specify their requirements in sufficient detail. Other clerks blame the suppliers for giving inaccurate or incomplete descriptions of their equipment in their catalogs. On the other hand, site engineers complain that they are not consulted when there are doubts regarding the choice of equipment.

This scenario basically describes one issue, namely that the equipment is being rejected upon delivery. We can see three primary causes from the issue, which are summarized in the cause–effect diagram in Fig. 6.2. The diagram also shows secondary causes underpinning each of the primary causes. Note that the factor "clerk selected equipment with incorrect specs" has been classified under the Material category because this factor stems from incorrect input data. A defect in input data used by a process falls under the Material category.

Exercise 6.2 Consider the university admission process described in Exercise 1.1 (p. 4). One of the issues faced by the university is that students have to wait too long to know the outcome of the application (especially for successful outcomes). It often happens that by the time a student is admitted, the student has decided to go to another university instead (students send multiple applications in parallel to many universities). Analyze the causes of this issue using a cause–effect diagram.

6.2.2 Why–Why Diagrams

Why–why diagrams (also known as *tree diagrams*) constitute another technique to analyze the cause of negative effects, such as issues in a business process. The emphasis of root cause analysis is to capture the series of cause-to-effect relations that lead to a given effect. The basic idea is to recursively ask the question: Why has something happened? This question is asked multiple times until a factor that stakeholders perceive to be a *root cause* is found. A common belief in the field of quality management—known as the five Why's principle—has it that answering the "why" question five times recursively allows one to pin down the root causes of a given negative effect. Of course, this should not be treated as gospel, but as a guideline of how far one should go during root cause analysis.

Why–why diagrams are a technique for structuring brainstorming sessions (e.g. workshops) for root cause analysis. Such a session would start with an issue. The first step is to give a name to the issue that stakeholders agree on. Sometimes it is found that there is not one issue, but multiple issues, in which case they should be analyzed separately. Once the issue has been identified and a name has been agreed upon, this becomes the root of the tree. Then at each level the following questions are asked: "Why does this happen?" and "What are the main sub-issues that may lead to this issue?". Possible factors are then identified. Each of these factors is then analyzed using the same questions. When getting down in the tree (e.g. to levels 3 or 4) it is recommended to start focusing on factors that can be resolved, meaning that something can be done to change them. The leaves of the tree should correspond to factors that are fundamental in nature, meaning that they cannot be explained in terms of other factors. Ideally, these factors, called root causes, should be such that they can be eliminated or mitigated, but this is not necessarily the case. For example, in the context of an insurance claims handling process, a certain type of errors in a police report may be due to lack of time and hectic schedules on the side of police agents involved in filling these reports. There is relatively little the insurance agency can do in this case to eliminate the error, other than raising the issue with the relevant authorities. Yet, the impact of this factor could be mitigated by putting in place checks to detect such errors as early as possible in the process.

A simple template for why–why diagrams is given in Fig. 6.3. An alternative way of presenting the information in such diagrams is by means of nested bullet-point lists. In the rest of this chapter we will opt for the latter representation.

Example 6.3 We consider again the equipment rental process described in Example 1.1 (p. 2). In Example 6.2 above, we noted that one of the issues with this process is that the site engineer sometimes rejected the equipment upon delivery because it was not suitable for the job at hand. Another issue is that BuildIT spends more in equipment rental than what it budgeted for. An auditor pointed out that one of the reasons for excessive expenditure was that site engineers were keeping the rented equipment longer than initially planned by using deadline extensions. Site engineers knew that it was easy to get a deadline extension. They also knew that it took quite some time to get equipment rental requests approved, and the larger the cost and

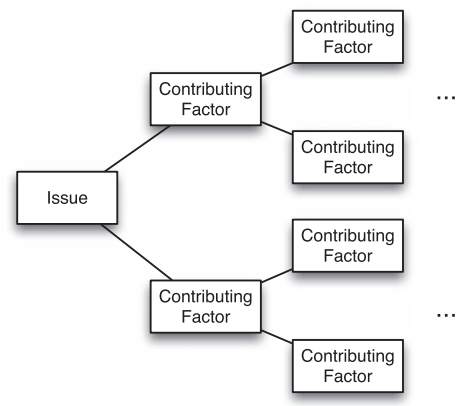


Fig. 6.3 Template of a why–why diagram

the duration of the rental, the slower it was to get it approved. So in many cases, site engineers were renting equipment several days before the date when they actually needed it. Also, they were specifying short periods in their equipment rental requests in order to get them approved quicker. When the deadline for returning an equipment approached, they just called the supplier to keep the equipment for a longer period.

Another issue spotted by the auditor is that a significant amount of late-payment penalty fees were paid to the suppliers because invoices for equipment rental were not paid by their due date. The clerks blamed the site engineers for being slow in approving the invoices.

In summary, we can distinguish at least three issues. First, the wrong equipment is being delivered on some occasions. Secondly site engineers are frequently asking for deadline extensions. Thirdly, BuildIT is often paying late payment fees to suppliers. A possible analysis root cause analysis of these issues leads to the following why–why diagrams (represented as nested bullet-point lists).

Issue 1 Site engineers sometimes reject delivered equipment, why?

Wrong equipment is delivered, why?

- miscommunication between site engineer and clerk, why?
  - site engineer provides only brief/inaccurate description of what they want
  - site engineer does not (always) see the supplier catalogs when making a request and does not communicate with the supplier, why?
    - · site engineer generally does not have Internet connectivity
  - site engineer does not check the choice of equipment made by the clerk
- equipment descriptions in supplier's catalog not accurate

Issue 2 Site engineers keep equipment longer than needed via deadline extensions, why?

Site engineer fears that equipment will not be available later when needed, why?

– time between request and delivery too long, why?
  - excessive time spent in finding a suitable equipment and approving the request, why?
    · time spent by clerk contacting possibly multiple suppliers sequentially
    · time spent waiting for works engineer to check the requests

Issue 3 BuildIT often has to pay late payment fees to suppliers, why?

Time between invoice received by clerk and confirmation is too long, why?

– clerk needs confirmation from site engineer, why?
  - clerk cannot assert when was the equipment delivered and picked-up, why?
    · delivery and pick-up of equipments are not recorded in a shared information system
    · site engineer can extend the equipment rental period without informing the clerk
  - site engineer takes too long to confirm the invoice, why?
    confirming invoices is not a priority for site engineer

Exercise 6.3 Consider again the process for university admission described in Exercise 1.1 (p. 4) and the issue described in Exercise 6.2 above. Analyze this issue using a why–why diagram.

6.3 Issue Documentation and Impact Assessment

Root cause analysis techniques allow us to understand the factors behind a given issue. A natural next step is to understand the impact of these issues. Building up this understanding is critical in order to prioritize the issues so that the attention of the process owner, participants and analysts can be focused on the issues that most matter to the organization. Below we discuss two complementary techniques for impact assessment.

6.3.1 Issue Register

The *issue register* complements the output of root cause analysis by providing a more detailed analysis of individual issues and their impact. The purpose of the issue register is to determine how and to what extent each issue is impacting on the performance of the process. The impact of an issue can be described quantitatively, for example in terms of time or money lost, or qualitatively, in terms of perceived nuisance to the customer or perceived risks that the issue entails. For example, nuisances caused to the customer because of misunderstandings during the execution of the process can be classified as qualitative impact, since it is difficult to translate this nuisance into a monetary measure.

Concretely, an issue register is a listing that provides a detailed analysis of each issue and its impact in the form of a table with a pre-defined set of fields. The following fields are typically described for each issue:

- *Name of the issue*. This name should be kept short, typically two–five words, and should be understandable by all stakeholders in the process.
- *Description*. A short description of the issue, typically one–three sentences, focused on the issue itself as opposed to its consequences or impact, which are described separately.
- *Priority*. A number $(1, 2, 3, \ldots)$ stating how important this issue is relative to other issues. Note that multiple issues can have the same priority number.
- *Assumptions* (or *input data*). Any data used or assumptions made in the estimation of the impact of the issue, such as for example number of times a given negative outcome occurs, or estimated loss per occurrence of a negative outcome. In the early phases of the development of the issue register, the numbers in this column will be mainly assumptions or ballpark estimates. Over time, these assumptions and rough estimates will be replaced with more reliable numbers derived from actual data about the execution of the process.
- *Qualitative impact*. A description of the impact of the issue in qualitative terms, such as impact of the issue on customer satisfaction, employee satisfaction, long-term supplier relationships, company's reputation or other intangible impact that is difficult to quantify.
- *Quantitative impact*. An estimate of the impact of the issue in quantitative terms, such as time loss, revenue loss or avoidable costs.

Other fields may be added to an issue register. For example, in view of process redesign, it may be useful to include an attribute *possible resolution* that describes possible mechanisms for addressing the issue.

Example 6.4 We consider again the equipment rental process described in Example 1.1 (p. 2) and the issues described above in Examples 6.2 and 6.3. The issue register given in Table 6.2 provides a more detailed analysis of these issues and their impact.[1]

Question Issue or factor?

An issue register is likely to contain a mixture of issues that have a direct impact on business performance, and others that are essentially causal or contributing factors of other issues that then impact on business performance. In other words, the issue register contains both issues and factors. For example, in the issue register of the equipment rental process, one could find the following entries:

- Clerk misunderstood the site engineer's requirements for an equipment.

[1] In this issue register we do not use multiple columns. This is a pragmatic choice to better fit the issue register within the width of the page.

Table 6.2 Issue register of equipment rental process

Issue 1: Equipment kept longer than needed
Priority: 1
Description: Site engineers keep the equipment longer than needed by means of deadline extensions
Assumptions: BuildIT rents 3000 pieces of equipment per year. In 10 % of cases, site engineers keep the equipment two days longer than needed to avoid disruptions due to delays in equipment rentals. On average, rented equipment costs € 100 per day
Qualitative impact: Not applicable
Quantitative impact: $0.1 \times 3000 \times 2 \times €\ 100 = €\ 60{,}000$ in additional rental expenses per year
Issue 2: Rejected equipment
Priority: 2
Description: Site engineers sometimes reject the delivered equipment due to non-conformance to their specifications
Assumptions: BuildIT rents 3000 pieces of equipment per year. Each time an equipment is rejected due to a mistake on BuildIT's side, BuildIT is billed the cost of one day of rental, that is € 100. 5 % of them are rejected due to an internal mistake within BuildIT (as opposed to a supplier mistake)
Qualitative impact: These events disrupt the construction schedules and create frustration and internal conflicts
Quantitative impact: $3000 \times 0.05 \times €\ 100 = €\ 15{,}000$ per year
Issue 3: Late payment fees
Priority: 3
Description: BuildIT pays late payment fees because invoices are not paid by the due date
Assumptions: BuildIT rents 3000 pieces of equipment per year. Each equipment is rented on average for 4 days at a rate of € 100 per day. Each rental leads to one invoice. About 10 % of invoices are paid late. On average, the penalty for late payment is 2 % of the amount of the invoice
Qualitative impact: Suppliers are annoyed and later unwilling to negotiate more favorable terms for equipment rental
Quantitative impact: $0.1 \times 3000 \times 4 \times €\ 100 \times 0.02 = €\ 2400$ per year

- Clerk did not select the correct equipment from the supplier's catalog due to inattention.
- Clerk indicated an incorrect delivery date in the PO and the supplier used this wrong date.
- Supplier did not deliver the exact equipment that had been ordered.
- Delivered equipment is faulty or is not ready-for-use.
- Supplier delivered the equipment to the wrong construction site or at the wrong time.

All of the above issues are possible causal or contributing factors of a top-level issue, namely "Equipment is rejected by the site engineer". The fact that the site engineer rejects the equipment creates a direct impact for BuildIT, for example in

terms of delays in the construction schedule. Meanwhile, the issues listed above have an indirect business impact, in the sense that they lead to the equipment being rejected and the needed equipment not being available on time, which in turn leads to delays in the construction schedule.

When an issue register contains a combination of issues and factors, it may be useful to add two fields to the register, namely "caused by" and "is cause of", that indicate for a given issue, which other issues in the register are related to it via a cause–effect relation. This way it becomes easier to identify which issues are related between them so that related issues can be analyzed together. Also, when an issue X is a factor of an issue Y, instead of analyzing both the impact of X and Y, we can analyze the impact of Y and in the qualitative and quantitative impact fields of X we can simply refer to the impact of Y. For example, in the impact field of issue "Clerk misunderstood the site engineer's requirements" we can simply refer to the impact of "Equipment is rejected by the site engineer".

Alternatively, we can adopt the convention of including in the issue register only top-level issues, meaning issues that have a direct business impact, and separately, we can use why–why diagrams and cause–effect diagrams to document the factors underpinning these top-level issues. This convention is followed in the rest of this chapter, meaning that the issue registers shown below only contain top-level issues rather than factors.

Exercise 6.4 Write an issue register for the university admission process and the issue described in Exercise 6.2.

6.3.2 Pareto Analysis and PICK Charts

The impact assessment conducted while building the issue register can serve as input for *Pareto analysis*. The aim of Pareto analysis is to identify which issues or which causal factors of an issue should be given priority. Pareto analysis rests on the principle that a small number of factors are responsible for the largest share of a given effect. In other words:

- A small subset of issues in the issue register are likely responsible for the largest share of impact.
- For a given issue, a small subset of factors behind this issue are likely responsible for the largest share of occurrences of this issue.

Sometimes this principle is also called the 80–20 principle, meaning that 20 % of issues are responsible for 80 % of the effect. One should keep in mind, however, that the specific proportions are only indicative. It may be for example that 30 % of issues are responsible for 70 % of the effect.

A typical approach to conduct Pareto analysis is as follows:

1. Define the effect to be analyzed and the measure via which this effect will be quantified. The measure might be for example:

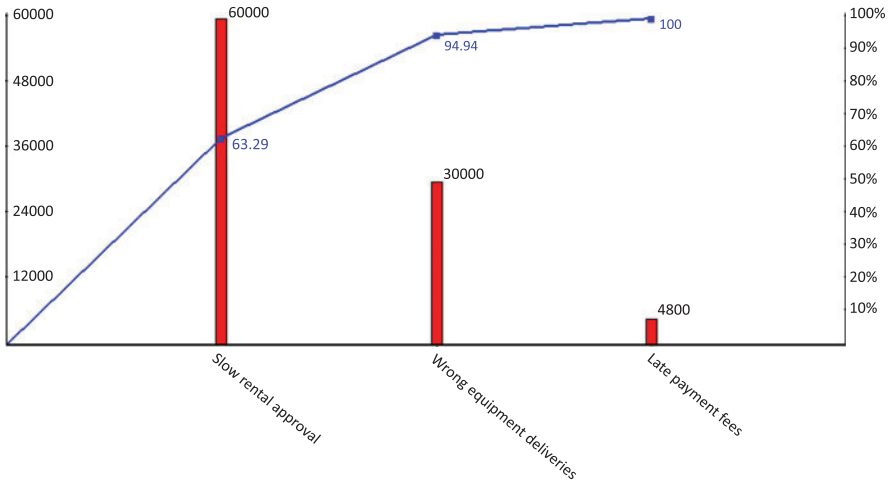


Fig. 6.4 Pareto chart for excessive equipment rental expenditure

- Financial loss for the customer or for the business.
- Time loss by the customer or by the process participants.
- Number of occurrences of a negative outcome, such as number of unsatisfied customers due to errors made when handling their case.

2. Identify all relevant issues that contribute to the effect to be analyzed.
3. Quantify each issue according to the chosen measure. This step can be done on the basis of the issue register, in particular, the quantitative impact column of the register.
4. Sort the issues according to the chosen measure (from highest to lowest impact) and draw a so-called *Pareto chart*. A Pareto chart consists of two components:
   a. A bar chart where each bar corresponds to an issue and the height of the bar is proportional to the impact of the issue or factor.
   b. A curve that plots the cumulative percentage impact of the issues. For example, if the issue with the highest impact is responsible for 40 % of the impact, this curve will have a point with a y-coordinate of 0.4 and an x-coordinate positioned so as to coincide with the first bar in the bar chart.

Example 6.5 Consider again the equipment rental process described in Example 1.1 (p. 2) and the issue register in Example 6.4. All three issues in this register share in common that they are responsible for unnecessary rental expenditure, which is a form of financial loss. From the data in the impact column of the register, we can plot the Pareto chart in Fig. 6.4.

This Pareto chart shows that issue "Slow rental approval" is responsible already for 63 % of unnecessary rental expenditure. Given that in this example there are only three issues, one could have come to this conclusion without conducting Pareto analysis. In practice though, an issue register may contain dozens or hundreds of issues, making Pareto analysis a useful tool to summarize the data in the issue register.

Exercise 6.5 Let us consider again the equipment rental process. This time we take the perspective of the site engineer, whose goal is to have the required equipment available on site when needed. From this perspective, the main issue is that in about 10 % of cases, the requested equipment is not available on site the day when it is required. When this happens, the site engineer contacts the suppliers directly to resolve the issue, but still, resolving the issue may take several days. It is estimated that each such delay costs € 400 per day to BuildIT. By inspecting a random sample of delayed equipment deliveries during a one-year period and investigating the cause of each occurrence, an analyst found that:

1. five occurrences were due to the site engineer not having ordered the equipment with sufficient advance notice: The site engineers ordered the equipment the day before it was needed, when at least two days are needed. These cases cause delays of one day on average.
2. nine occurrences were due to the fact that none of BuiltIT's suppliers had the required type of equipment available on the requested day. These cases cause delays of one to four days (three days on average).
3. 13 occurrences were due to the approval process taking too long (more than a day) due to mistakes or misunderstandings. For these cases, the delay was one day on average.
4. 27 occurrences were due to the equipment having been delivered on time, but the equipment was not suitable and the site engineer rejected it. These cases cause delays of two days on average.
5. four occurrences were due to mistakes or delays attributable entirely to the supplier. These cases lead to delays of one day. However, in these cases, the supplier compensated BuildIT by providing the equipment two days for free (the remaining days are still charged). Recall that the average cost of an equipment rental per day is € 100.
6. For two occurrences, the analyst did not manage to determine the cause of the delay (the process participants could not recall the details). The delays in these cases where two days per occurrence.

The sample of analyzed occurrences represents around 20 % of all occurrences of the issue during a one-year period.

Draw a Pareto chart corresponding to the above data.

It is worth highlighting that Pareto analysis focuses on a single dimension. In the example above, the dimension under analysis is the impact in monetary terms. In other words, we focus on the estimated payoff of addressing an issue. In addition to payoff, there is another dimension that should be taken into account when deciding which issues should be given higher priority, namely the level of difficulty of addressing an issue. This level of difficulty can be quantified in terms of the amount of investment required to change the process in order to address the issue in question.

A type of chart that can be used as a complement to Pareto charts in order to take into account the difficulty dimension is the *PICK chart*. A PICK chart (see Fig. 6.5) is a four-quadrant chart where each issue appears as a point. The horizontal axis

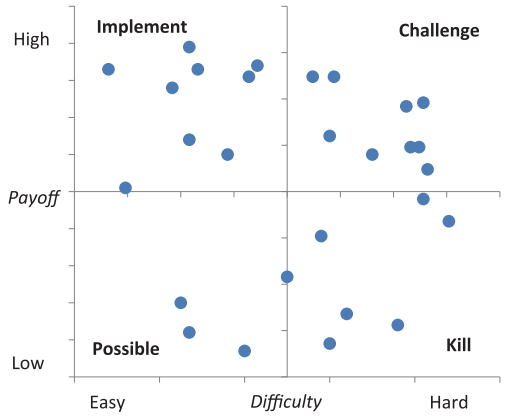


Fig. 6.5 PICK chart

captures the difficulty of addressing the issue (or more specifically the difficulty of implementing a given improvement idea that addresses the issue) while the vertical axis captures the payoff. The horizontal axis (difficulty) is split into two sections (easy and hard) while the vertical axis (payoff) is split into low and high. These splits lead to four quadrants that allow analysts to classify issues according to the trade-off between payoff and difficulty:

- *Possible* (low payoff, easy to do): issues that can be addressed if there are sufficient resources for doing so.
- *Implement* (high payoff, easy to do): issues that should definitely be implemented as a matter of priority.
- *Challenge* (high payoff, hard to do): issues that should be addressed but require significant amount of effort. In general one would pick one of these challenges and focus on it rather than addressing all or multiple challenges at once.
- *Kill* (low payoff, hard to do): issues that are probably not worth addressing or at least not to their full extent.

6.4 Recap

In this chapter, we presented a selection of techniques for qualitative analysis of business processes. The first presented technique, namely value-added analysis, aims at identifying waste, specifically time wasted in activities that do not give value to the customer or to the business. Next, we presented two techniques to uncover the causes of issues that affect the performance of a process, namely cause–effect analysis and why–why analysis. Whereas cause–effect analysis focuses on classifying the factors underpinning the occurrences of an issue, why–why analysis focuses on identifying the recursive cause–effect relations between these factors.

Finally, we presented an approach to systematically document issues in a process, namely the issue register. The purpose of an issue register is to document issues

in a semi-structured way and to analyze their impact on the business both from a qualitative and a quantitative angle. In particular, the issue register provides a starting point to build Pareto charts and PICK charts—two visualization techniques that provide a bird's-eye view of a set of issues. These charts help analysts to focus their attention on issues that offer the best payoff (in the case of Pareto charts) or the best trade-off between payoff and difficulty (in the case of PICK charts).

6.5 Solutions to Exercises

Solution 6.1

- VA: receive online application, evaluate academic admissibility, send notification to student.
- BVA: check completeness, academic recognition agency check, English test check.
- NVA: receive physical documents from students, forward documents to committee, notify students service of outcomes of academic admissibility.

Note In this solution we treat the entire agency check as BVA. Part of this agency check consists of the admissions office sending the documents to the agency and the agency sending back the documents and their assessment to the admissions office. These two sub-steps could be treated as NVA. However, if we assume that the agency requires the documents to be sent by post to them, these sub-steps cannot be easily separated from the agency check itself. In other words, it would not be possible to eliminate these handover steps without eliminating the entire agency check. Thus the entire agency check should arguably be treated as a single step.

Solution 6.2 The cause–effect diagram corresponding to this exercise should include at least the name of the issue (e.g. "Student waiting time too long") and the following factors:

- Process stalls due to agency check. This is a "Method" issue, since the issue stems from the fact that the process essentially stalls until a response is received from the agency. One could argue that to some extent this is a "Milieu" issue. But while the slowness of the agency check is a "Milieu" issue, the fact that the process stalls until a response is received from the agency is a "Method" issue.
- Agency check takes too long. This is a "Milieu" issue since the agency is a separate entity that imposes its own limitations.
- Academic committee assessment takes too long. This is a "Method" issue since the process imposes that the academic committee only assesses applications at certain times (when it meets), rather than when applications are ready to be evaluated.
- Physical documents take too long to be received. This is a "Milieu" issue for two reasons. First, the physical documents are needed for the purpose of the agency

check and the delays in the arrival of physical documents are caused by the applicants themselves and postal service delays.
- Admission office delays the notification after academic assessment. This seems to be a "Method" issue, but the description of the process does not give us sufficient information to state this conclusively. Here, a process analyst would need to gather more information in order to understand this issue in further detail.

Solution 6.3
Admission process takes too long, why?

– Process stalls until physical documents arrive, why?
  - Agency check requires physical documents.
  - Other tasks are performed only after agency check, why?
    · Traditionally this is how the process is set-up but there is no strong reason for it.
– Agency check takes too long, why?
  - Exchanges with the agency are via post, why?
    · Agency requires original (or certified) documents due to regulatory requirements.

Academic committee takes too long, why?

– Documents are exchanged by internal mail between admissions office and committee.
– Academic committee only meets at specified times.

Admission office delays the notification after academic assessment, why?

– Not enough information available to analyze this issue (probably due to batching —admissions office sends notifications in batches).

The above analysis already suggests one obvious improvement idea: perform the academic committee assessment in parallel to the agency check. Another improvement opportunity is to replace internal mail communication between admissions office and academic committee with electronic communication (e.g. documents made available to committee members via a Web application).

Note that we could have done the analysis starting from the issue "Admitted students reject their admission offer". This might be useful since there might be several reasons why students reject their offer, some related to the admission process, but also some unrelated to the process.

Solution 6.4 In the following issue register, we only analyze the issue described in this chapter, namely that the admission process takes too long. In practice, the issue register would include multiple issues.

Issue 1: Students reject offer due to long waiting times
Priority: 1
Description: The time between online submission of an application to notification of acceptance takes too long, resulting in some students rejecting their admission offer
Assumptions: Circa 20 students per admission round reject their offer because of the delays. Assessment of each application costs € 100 per student to the university in time spent by admissions office and academic committee, plus an additional € 50 for the agency check. University spends € 100 in marketing for each application it attracts
Qualitative impact: Students who would contribute to the institution in a positive way are lost. Delays in the admission process affect the image of the university vis-a-vis of future students, and generate additional effort to handle enquiries from students while they wait for the admission decisions
Quantitative impact: $20 \times$ € $250 =$ € 5000 per admission round

In the above issue analysis, the effort required to deal with enquiries during the pre-admission period is listed in the qualitative impact field. If it was possible (with a reasonable amount of effort) to estimate how many such enquiries arrive and how much time they consume, it would be possible to turn this qualitative impact into a quantitative one.

Solution 6.5 First, we analyze the cost incurred by each type of occurrence (i.e. each causal factor) in the sample:

1. Last-minute request: one day delay (because normally two days advance notice are needed), thus € 400 cost $\times 5 =$ € 2000.
2. Equipment out-of-stock: three days delay $=$ € $1200 \times 9 =$ € 10,800.
3. Approval delay: one day delay $=$ € $400 \times 13 =$ € 5200.
4. Rejected equipment: two days delay $=$ € $800 \times 27 =$ € 21,600. Note that in Example 6.4 we mentioned that when an equipment is rejected, a fee of € 100 (on average) has to be paid to the supplier for taking back the equipment. However, we do not include this fee here because we are interested in analyzing the costs stemming from equipment not being available on the required day, as opposed to other costs incurred by rejecting equipments.
5. Supplier mistake: one day delay $=$ € 400 minus € 200 in rental cost saving $=$ € $200 \times 4 =$ € 800.
6. Undetermined: two days delay $=$ € $800 \times 2 =$ € 1600.

Since the sample represents 20 % of occurrences of the issue over a year, we multiply the above numbers by five in order to estimate the total yearly loss attributable to each causal factor. The resulting Pareto chart is given in Fig. 6.6.

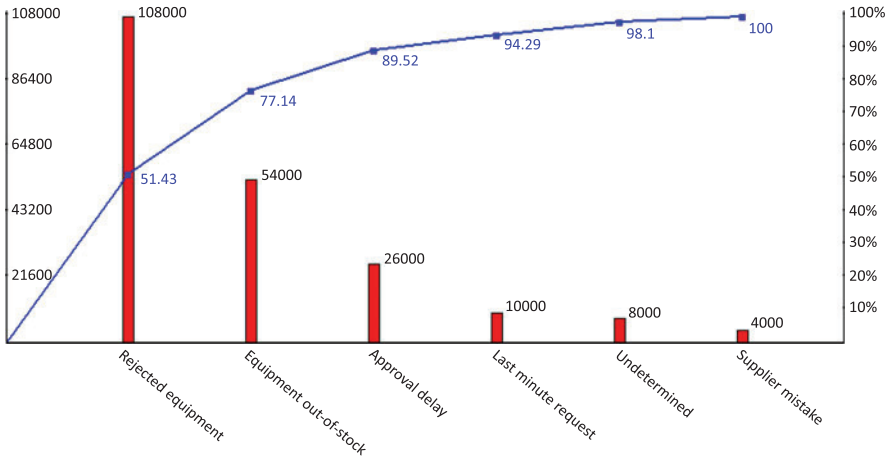


Fig. 6.6 Pareto chart of causal factors of issue "Equipment not available when needed"

6.6 Further Exercises

Exercise 6.6 Consider the following summary of issues reported in a travel agency.

> A travel agency has recently lost several medium-sized and large corporate customers due to complaints about poor customer service. The management team of the travel agency decided to appoint a team of analysts to address this problem. The team gathered data by conducting interviews and surveys with current and past corporate customers and also by gathering customer feedback data that the travel agency has recorded over time. About 2 % of customers complained about errors that had been made in their bookings. In one occasion, a customer had requested a change to a flight booking. The travel agent wrote an e-mail to the customer suggesting that the change had been made and attached a modified travel itinerary. However, it later turned out that the modified booking had not been confirmed in the flight reservation system. As a result, the customer was not allowed to board the flight and this led to a series of severe inconveniences for the customer. Similar problems had occurred when booking a flight initially: the customer had asked for certain dates, but the flight tickets had been issued for different dates. Additionally, customers complained of the long times it took to get responses to their requests for quotes and itineraries. In most cases, employees of the travel agency replied to requests for quotes within 2–4 working hours, but in the case of some complicated itinerary requests (about 10 % of the requests), it took them up to 2 days. Finally, about 5 % of customers also complained that the travel agents did not find the best flight connections and prices for them. These customers essentially stated that they had found better itineraries and prices on the Web by searching by themselves.

1. Analyze the issues described above using root cause analysis techniques.
2. Document the issues in the form of an issue register. To this end, you may assume that the travel agency receives around 100 itinerary requests per day and that the agency makes 50 bookings per day. Each booking brings a gross profit of € 100 to the agency.

Exercise 6.7 Consider the pharmacy prescription fulfillment process described in Exercise 1.6 (p. 28). Identify the steps in this process and classify them into value-adding, business value-adding and non-value-adding.

Exercise 6.8 Consider the procure-to-pay process described in Exercise 1.7 (p. 29). Identify the steps in this process and classify them into value-adding, business value-adding and non-value-adding.

Exercise 6.9 Write an issue register for the pharmacy prescription fulfillment process described in Exercise 1.6 (p. 28). Analyze at least the following issues:

- Sometimes, a prescription cannot be filled because one or more drugs in the prescription are not in stock. The customer only learns this when they come to pick up their prescription.
- Oftentimes, when the customer arrives to pick up the drugs, they find out that they have to pay more than what they expected because their insurance policy does not cover the drugs in the prescription, or because the insurance company covers only a small percentage of the cost of the drugs.
- In a very small number of cases, the prescription cannot be filled because there is a potentially dangerous interaction between one of the drugs in the prescription and other drugs that the customer has been given in the past. The customer only finds out about this issue when they arrive to pick up the prescription.
- Some prescriptions can be filled multiple times. This is call a "refill". Every prescription explicitly states whether a refill is allowed and if so how many refills are allowed. Sometimes, a prescription cannot be filled because the number of allowed refills has been reached. The pharmacist then tries to call the doctor who issued the prescription to check if the doctor would allow an additional refill. Sometimes, however, the doctor is unreachable or the doctor does not authorize the refill. The prescription is then left unfilled and the customer only finds it out when they arrive to pick-up the prescription.
- Oftentimes, especially during peak time, customers have to wait for more than 10 minutes to pick-up their prescription due to queues. Customers find this annoying because they find that having to come twice to the pharmacy (once for drop-off and once for pick-up) should allow the pharmacy ample time to avoid such queues at pick-up.
- Sometimes, the customer arrives at the scheduled time, but the prescription is not yet filled due to delays in the prescription fulfillment process.

When making assumptions to analyze these issues, you may choose to equate "oftentimes" with "20 % of prescriptions", "sometimes" with "5 % of prescriptions" and "very small number of cases" with "1 % of prescriptions". You may also assume that the entire chain of pharmacies consists of 200 pharmacies that serve 4 million prescriptions a year and that the annual revenue of the pharmacy chain attributable to prescriptions is € 200 million. You may also assume that every time a customer is dissatisfied when picking up a prescription, the probability that this customer will

not come back after this experience is 20 %. You may also assume that on average a customer requires five prescriptions per year.

Taking the issue register as a basis, apply Pareto Analysis to determine a subset of issues that should be addressed to reduce the customer churn due to dissatisfaction by at least 70 %. Customer churn is the number of customers who stop consuming services offered by a company at a given point in time. In this context, this means the number of customers who stop coming to the pharmacy due to a bad customer experience.

Exercise 6.10 Write an issue register for the procure-to-pay process described in Exercise 1.7 (p. 29).

6.7 Further Reading

Value-added analysis, cause–effect analysis, why–why analysis and Pareto analysis are just a handful of a much wider range of techniques used in the field of Six Sigma (cf. "Related Fields" box in Chap. 1). Conger [8] shows how these and other Six Sigma techniques can be applied for business process analysis. The list of analysis techniques encompassed by Six Sigma is very extensive. A comprehensive listing of Six Sigma techniques is maintained in the iSixSigma portal (http://www.isixsigma.com/tools-templates/). A given business process improvement project will generally only make use of a subset of these techniques. In this respect, Johannsen et al. [38] provide guidelines for selecting analysis techniques for a given BPM project.

Straker's Quality Encyclopedia (see http://www.syque.com/improvement/a_encyclopedia.htm) provides a comprehensive compendium of concepts used in Six Sigma and other quality management disciplines. In particular, it provides definitions and illustrations of the 6M's and the 4P's used in cause–effect diagrams and other concepts related to root cause analysis. A related resource—also by Straker—is the Quality Toolbook, which summarizes a number of quality management techniques. Originally the Quality Toolbook was published as a hardcopy book [89], but it is nowadays also available freely in hyperlinked form at: http://www.syque.com/quality_tools/toolbook/toolbook.htm.

Why–why diagrams allow us to document sequences of cause–effect relations that link factors to a given issue. A related technique to capture cause–effect paths is the *causal factor chart* [77]. Causal factor charts are similar to why–why diagrams. A key difference is that in addition to capturing factors, causal factor charts also capture conditions surrounding the factors. For example, in addition to stating that "the clerk made a data entry mistake when creating the PO", a casual factor chart might also include a condition corresponding to the question "in which part of the PO the clerk made a mistake?" These additional conditions allow analysts to more clearly define each factor.

The issue register has been proposed as a process analysis tool by Schwegmann and Laske [84][2] who use the longer term "list of weaknesses and potential improvements" to refer to an issue register. Schwegmann and Laske argue that the issue register should be built up in parallel with the as-is model, meaning that the discovery of the as-is process and the documentation of issues should go hand in hand. The rationale is that during the workshops organized for the purpose of process discovery (cf. Chap. 5), workshop participants will often feel compelled to voice out issues related to different parts of the process. Therefore, process discovery is an occasion to start listing issues. Naturally, during process discovery, the documentation of issues is left incomplete because the focus is more on understanding the as-is process. Additional analysis after the process discovery phase is required in order to document the issues and their impact in detail.

Another framework commonly used for qualitative process analysis is the Theory of Constraints (TOC) [23]. TOC is especially useful when the goal is to trace weaknesses in the process to specific bottlenecks. In addition to providing a framework for process analysis, TOC also provides guidance to identify, plan and implement changes in order to address the identified bottlenecks. The application of TOC to business process analysis and redesign is discussed at length by Laguna and Marklund [43, Chap. 5] and by Rhee et al. [76].

Finally, a useful framework when it comes to analyzing individual tasks (or activities) in a business process—as opposed to the entire process—is provided by Harmon [31, Chap. 10].[3] This so-called Task Analysis or Activity Analysis framework includes a comprehensive collection of questions and checklists that an analyst should answer and complete in order to identify opportunities to improve the performance of a given activity.

[2]The sub-categorization of the 6M's given in Sect. 6.2.1 also comes from Schwegmann and Laske [84].

[3]An alternate reference describing this framework is [32].

Chapter 7
Quantitative Process Analysis

It is better to be approximately right than precisely wrong.
Warren Buffett (1930–)

Qualitative analysis is a valuable tool to gain systematic insights into a process. However, the results obtained from qualitative analysis are sometimes not detailed enough to provide a solid basis for decision making. Think of the process owner of BuildIT's equipment rental process preparing to make a case to the company's COO that every site engineer should be given a tablet computer with wireless access in order to query suppliers' catalogs and to make rental requests from any construction site. The process owner will be asked to substantiate this investment in quantitative terms and specifically to estimate how much time and money would be saved by doing this investment. To make such estimates, we need to go beyond qualitative analysis.

This chapter introduces a range of techniques for analyzing business processes quantitatively, in terms of performance measures such as cycle time, total waiting time and cost. Specifically, the chapter focuses on three techniques: flow analysis, queueing analysis and simulation. All these techniques have in common that they allow us to calculate performance measures of a process, given data about the performance of individual activities and resources in the process.

7.1 Performance Measures

7.1.1 Process Performance Dimensions

Any company would ideally like to make its processes faster, cheaper, and better. This simple observation leads us already to identifying three *process performance dimensions*: time, cost and quality. A fourth dimension gets involved in the equation once we consider the issue of change. A process might perform extremely well under normal circumstances, but then perform poorly in other circumstances which are perhaps equally or more important. For example, van der Aalst et al. [98] report the story of a business process for handling claims at an Australian insurance

M. Dumas et al., *Fundamentals of Business Process Management*,
DOI 10.1007/978-3-642-33143-5_7,

company. Under normal, everyday conditions, the process was performing to the entire satisfaction of all managers concerned (including the process owner). However, Australia is prone to storms and some of these storms cause serial damages to different types of properties (e.g. houses and cars), leading to numerous claims being lodged in a short period of time. The call center agents and backoffice workers involved in the process were over-flooded with claims and the performance of the process degraded—precisely at the time when the customers were most sensitive to this performance. What was needed was not to make the process faster, cheaper or better during normal periods. Rather, it was needed to make the process more flexible to sudden changes in the amount of claims. This observation leads us to identify a fourth dimension of process performance, namely flexibility.

Each of the four performance dimensions mentioned above (time, cost, quality, and flexibility) can be refined into a number of *process performance measures* (also called *key performance indicators* or *KPIs*). A process performance measure is a quantity that can be unambiguously determined for a given business process—assuming of course that the data to calculate this performance measure is available.

For example, there are several types of cost such as cost of production, cost of delivery or cost of human resources. Each of these types of cost can be further refined into specific performance measures. To do so, one needs to select an aggregation function, such as count, average, variance, percentile, minimum, maximum, or ratios of these aggregation functions. A specific example of a cost performance measure is the average delivery cost per item.

Below, we briefly discuss each of the four dimensions and how they are typically refined into specific performance measures.

Time Often the first performance dimension that comes to mind when analyzing processes is time. Specifically, a very common performance measure for processes is *cycle time* (also called *throughput time*). Cycle time is the time that it takes to handle one case from start to end. Although it is usually the aim of a redesign effort to reduce cycle time, there are many different ways of further specifying this aim. For example, one can aim at a reduction of the average cycle time or the maximal cycle time. It is also possible to focus on the ability to meet cycle times that are agreed upon with a client at run time. Yet another way of looking at cycle time is to focus on its variation, which is notably behind approaches like Six Sigma (cf. Chap. 1). Other aspects of the time dimension come into view when we consider the constituents of cycle time, namely:

1. *Processing time* (also called *service time*): the time that resources (e.g. process participants or software applications invoked by the process) spend on actually handling the case.
2. *Waiting time*: the time that a case spends in idle mode. Waiting time includes *queueing time*—waiting time due to the fact that no resources available to handle the case—and other waiting time, for example because synchronization must take place with another process or because input is expected from a customer or from another external actor.

Cost Another common performance dimension when analyzing and redesigning a business process has a financial nature. While we refer to cost here, it would also have been possible to put the emphasis on turnover, yield, or revenue. Obviously, a yield increase may have the same effect on an organization's profit as a decrease of cost. However, process redesign is more often associated with reducing cost. There are different perspectives on cost. In the first place, it is possible to distinguish between fixed and variable cost. Fixed costs are overhead costs which are (nearly) not affected by the intensity of processing. Typical fixed costs follow from the use of infrastructure and the maintenance of information systems. Variable cost is positively correlated with some variable quantity, such as the level of sales, the number of purchased goods, the number of new hires, etc. A cost notion which is closely related to productivity is *operational cost*. Operational costs can be directly related to the outputs of a business process. A substantial part of operational cost is usually labor cost, the cost related to human resources in producing a good or delivering a service. Within process redesign efforts, it is very common to focus on reducing operation cost, particularly labor cost. The automation of tasks is often seen as an alternative for labor. Obviously, although automation may reduce labor cost, it may cause incidental cost involved with developing the respective application and fixed maintenance cost for the life time of the application.

Quality The quality of a business process can be viewed from at least two different angles: from the client's side and from the process participant's side. This is also known as the distinction between external quality and internal quality. The *external quality* can be measured as the client's satisfaction with either the product or the process. Satisfaction with the product can be expressed as the extent to which a client feels that the specifications or expectations are met by the delivered product. On the other hand, a client's satisfaction with the process concerns the way how it is executed. A typical issue is the amount, relevance, quality, and timeliness of the information that a client receives during execution on the progress being made. On the other hand, the *internal quality* of a business process related to the process participants' viewpoint. Typical internal quality concerns are: the level that a process participants feels in control of the work performed, the level of variation experienced, and whether working within the context of the business process is felt as challenging. It is interesting to note that there are various direct relations between the quality and other dimensions. For example, the external process quality is often measured in terms of time, e.g., the average cycle time or the percentage of cases where deadlines are missed. In this book, we make the choice that whenever a performance measure refers to time, it is classified under the time dimension even if the measure is also related to quality.

Flexibility The criterion that is least noted to measure the effect of a redesign measure is the flexibility of a business process. Flexibility can be defined in general terms as the ability to react to changes. These changes may concern various parts of the business process, for example:

- The ability of resources to execute different tasks within a business process setting.
- The ability of a business process as a whole to handle various cases and changing workloads.
- The ability of the management in charge to change the used structure and allocation rules.
- The organization's ability to change the structure and responsiveness of the business process to wishes of the market and business partners.

Another way of approaching the performance dimension of flexibility is to distinguish between run time and build time flexibility. *Run time flexibility* concerns the opportunities to handle changes and variations while executing a specific business process. *Build time flexibility* concerns the possibility to change the business process structure. It is increasingly important to distinguish the flexibility of a business process from the other dimensions.

Example 7.1 Let us consider the following scenario.

> A restaurant has recently lost many customers due to poor customer service. The management team has decided to address this issue first of all by focusing on the delivery of meals. The team gathered data by asking customers about how quickly they liked to receive their meals and what they considered as an acceptable wait. The data suggested that half of the customers would prefer their meals to be served in 15 minutes or less. All customers agreed that a waiting time of 30 minutes or more is unacceptable.

In this scenario, it appears that the most relevant performance dimension is time, specifically serving time. One objective that distills from the scenario is to completely avoid waiting times above 30 minutes. In other words, the percentage of customers served in less than 30 minutes should be as close as possible to 100 %. Thus, "percentage of customers served in less than 30 minutes" is a relevant performance measure. Another threshold mentioned in the scenario is 15 minutes. There is a choice between aiming to have an average meal serving time below 15 minutes or again, minimizing the number of meals served above 15 minutes. In other words, there is a choice between two performance measures: "average meal delivery time" or "percentage of customers served in 15 minutes".

This example illustrates that the definition of performance measures is tightly connected to the definition of *performance objectives*. In this respect, one possible method for deriving performance measures for a given process is the following:

1. Formulate performance objectives of the process at a high level, in the form of a desirable state that the process should ideally reach, e.g. "customers should be served in less than 30 minutes".
2. For each performance objective, identify the relevant performance dimension(s) and aggregation function(s), and from there, define one or more performance measures for the objective in question, e.g. "percentage of customers served in less than 30 minutes". Let us call this measure ST_{30}.
3. Define a more refined objective based on this performance measure, such as $ST_{30} \geq 99\ \%$.

During the redesign and implementation phases, a possible additional step is to attach a timeframe to the refined performance objective. For example, one can state that the above performance objective should be achieved in 12 months time. A performance objective with a timeframe associated to it is usually called a *performance target*. At the end of the chosen timeframe, one can assess to what extent the redesigned process has attained its targets.

Exercise 7.1 Consider the travel agency scenario described in Exercise 6.6 (p. 208).

1. Which business processes should the travel agency improve?
2. For each of the business processes you identified above, indicate which performance measure should the travel agency improve.

7.1.2 Balanced Scorecard

Another way of classifying and defining performance measures is given by the concept of *Balanced Scorecard*. The Balanced Scorecard is essentially an approach to align the goals and measures that are used to evaluate the work of managers. The main argument behind the Balanced Scorecard is that it is not sufficient to use financial metrics, such as Return-On-Investment (ROI) or operating margin, when evaluating managers. An extreme focus on these measures is in the long-term detriment to the company as it neglects fundamental sources of value, namely the customer, the company's internal structure and the company's employees. Accordingly, the Balanced Scorecard is based on four performance dimensions—each one covering a fundamental concern of a company:

- Financial Measures, e.g. cash flow, to ensure survival, operating margin to ensure shareholder satisfaction.
- Internal Business Measures, e.g. cycle time, to ensure efficiency and low levels of inventory in the case of manufacturing organizations.
- Innovation and Learning Measures, e.g. technology leadership, to ensure competitive advantage and to attract and retain talent.
- Customer Measures, e.g. on-time delivery, to ensure customer satisfaction and loyalty.

The classical way to implement the Balanced Scorecard follows a top-down procedure. It begins with a corporate scorecard, followed by departmental ones with an emphasis on goals and metrics directly affected by the specific department. Process-related measures tend to appear only at the level of heads of units or their subordinates. This classical implementation of the Balanced Scorecard overemphasizes the functional division of organizations not paying enough attention to processes view. Companies implementing the Balanced Scorecard in conjunction with BPM need to carefully consider the relation between the measures in the Balanced Scorecard—both at the corporate level, departmental level and lower levels—and the performance measures associated with their business processes. One way to ensure this

alignment is to implement a Balanced Scorecard structured according to the company's process architecture (cf. Chap. 2). This process-oriented Balanced Scorecard may co-exist with a Balanced Scorecard that is traditionally associated to the company's functional architecture.

In any case, we observe that the Balanced Scorecard is a useful tool for identifying process performance measures across an entire organization. This is in contrast with the method based on performance dimensions outlined in Sect. 7.1.1 is geared towards identifying performance measures for one given process. Thus, this latter method and the Balanced Scorecard are complementary.

7.1.3 Reference Models and Industry Benchmarks

Reference process models—previously mentioned in Chap. 2—provide another basis to identify process performance measures. For instance, within the Supply Chain Operations Reference Model (SCOR), processes and activities in the process hierarchy are linked to performance measures. An example of a performance measure in SCOR is the "Purchase Order Cycle Time", defined as the "average amount of time (e.g. days) between the moment an intention to purchase is declared and the moment the corresponding purchase order is received by the relevant vendor". This is basically the average cycle time of a fragment of a procure-to-pay process. Other measures in SCOR deal with inventory management or out-of-stock events. In addition to defining performance measures, SCOR also provides threshold values for each measure that allow a company to compare itself against peers within its industry and to determine whether they are in the top-10 %, top-50 % or bottom-50 % with respect to other companies in their industry sector.

Another relevant framework mentioned in Chap. 2 is APQC's Process Classification Framework (PCF). The primary aim of this framework is to provide a standardized decomposition of processes in an organization together with standardized names and definitions for these processes. As a complement to PCF, APQC has also developed a set of performance measures for the processes included in PCF. This is also a potentially useful tool for performance measure identification.

Yet another example of a reference model that provides a catalog of process performance measures is the *IT Infrastructure Library*—ITIL. ITIL's performance measures include, for example, "Incidents due to Capacity Shortages" defined as the "number of incidents occurring because of insufficient service or component capacity". This performance measure is linked to ITIL's Capacity Management process area, which includes a number of inter-related processes to manage the capacity of IT processes or components of an IT system.

Other reference models that provide catalogs of process performance measures include DCOR (Design Chain Operations Reference model) and eTOM (Enhanced Telecom Operations Map).

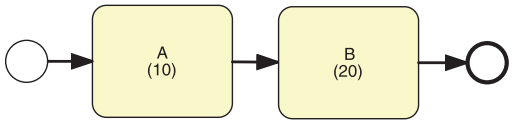


Fig. 7.1 Fully sequential process model

7.2 Flow Analysis

Flow analysis is a family of techniques that allow us to estimate the overall performance of a process given some knowledge about the performance of its activities. For example, using flow analysis we can calculate the average cycle time of an entire process if we know the average cycle time of each activity. We can also use flow analysis to calculate the average cost of a process instance knowing the cost-per-execution of each activity, or calculate the error rate of a process given the error rate of each activity.

In order to understand the scope and applicability of flow analysis, we start by showing how flow analysis can be used to calculate the average cycle time of a process. As a shorthand, we will use the term *cycle time* to refer to *average cycle time* in the rest of this chapter.

7.2.1 Calculating Cycle Time Using Flow Analysis

We recall that the cycle time of a process is the average time it takes between the moment the process starts and the moment it completes. By extension, we say that the cycle time of an activity is the average time it takes between the moment the activity is ready to be executed and the moment it completes.

To understand how flow analysis works, it is useful to start with an example of a purely sequential process as in Fig. 7.1. The cycle time of each activity is indicated between brackets. Since the two activities in this process are performed one after the other, we can intuitively conclude that the cycle time of this process is $20 + 10 = 30$. More generally, it is quite intuitive that the cycle time of a purely sequential fragment of a process is the sum of the cycle times of the activities in the fragment.

When a process model or a fragment of a model contains gateways, the cycle time is no longer the sum of the activity cycle times. Let us consider the example shown in Fig. 7.2. Here, it is clear that the cycle time of the process is not 40 (the sum of the activity cycle times). Indeed, in a given instance of this process, either activity B or activity C is performed. If B is performed, the cycle time is 30, while if C is performed, the cycle time is 20.

Whether the cycle time of this process is closer to 20 or closer to 30 depends on how frequently each branch of the XOR-split is taken. For instance, if in 50 % of instances the upper branch is taken and the remaining 50 % of instances the lower branch is taken, the overall cycle time of the process is 25. However, if the

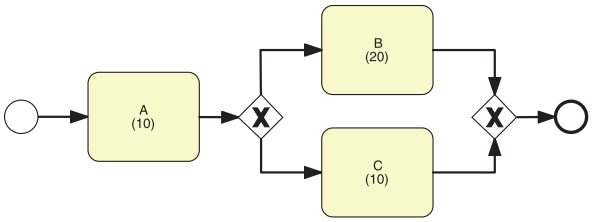


Fig. 7.2 Process model with XOR-block

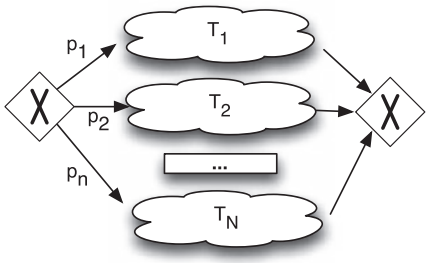


Fig. 7.3 XOR-block pattern

upper branch is taken only 10 % of the times and the lower branch is taken 90 % of the times, the cycle time should be intuitively closer to 30. Generally speaking, the cycle time of the fragment of the process between the XOR-split and the XOR-join is the weighted average of the cycle times of the branches in-between. Thus, if the lower branch has a frequency of 10 % and the upper branch has a frequency of 90 %, the cycle time of the fragment between the XOR-split and the XOR-join is: $0.1 \times 10 + 0.9 \times 20 = 19$. We then need to add the cycle time of activity A (which is always executed) in order to obtain the total cycle time, that is, $10 + 19 = 29$. In the rest of this chapter, we will use the term *branching probability* to denote the frequency with which a given branch of a decision gateway is taken.

In more general terms, the cycle time of a fragment of a process model with the structure shown in Fig. 7.3 is

$$CT = \sum_{i=1}^{n} p_i \times T_i \tag{7.1}$$

In Fig. 7.3, p_1, p_2, etc. are the branching probabilities. Each "cloud" represents a fragment that has a single entry flow and a single exit flow. The cycle times of these nested fragments are T_1, T_2, etc. Hereon, this type of fragment is called a *XOR-block*.

Let us now consider the case where parallel gateways are involved as illustrated in Fig. 7.4.

Again, we can observe that the cycle time of this process cannot be 40 (the sum of the activity cycle times). Instead, since tasks B and C are executed in parallel, their combined cycle time is determined by the slowest of the two activities, that is,

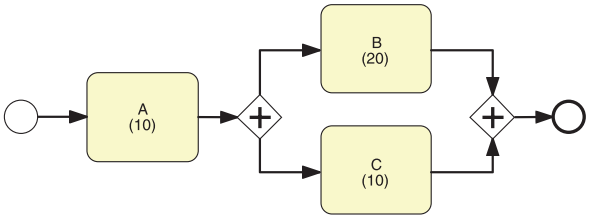


Fig. 7.4 Process model with AND-block

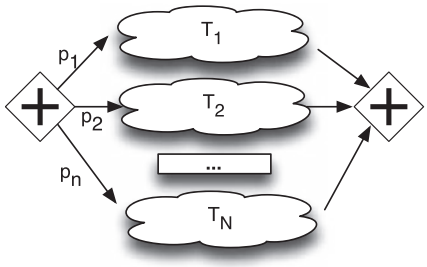


Fig. 7.5 AND-block pattern

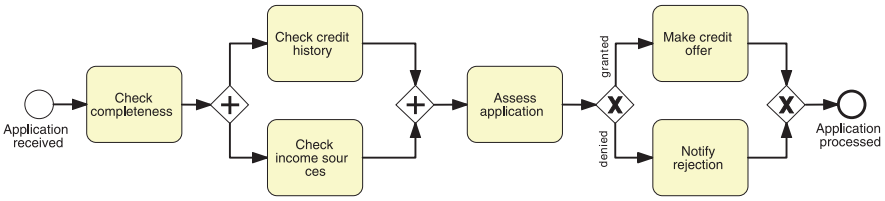


Fig. 7.6 Credit application process

by C. Thus, the cycle time of the process shown in Fig. 7.4 is $10 + 20 = 30$. More generally, the cycle time of an *AND-block* such as the one shown in Fig. 7.5 is

$$CT = Max(T_1, T_2, \ldots, T_n) \tag{7.2}$$

Example 7.2 Let us consider the credit application process model in Fig. 7.6 and the activity cycle times given in Table 7.1. Let us also assume that in 60 % of cases, the credit is granted.

To calculate the cycle time of this process, we first note that the cycle time of the AND-block is 3 days (slowest activity). Next, we calculate the cycle time of the fragment between the XOR-block using (7.1), that is, $0.6 \times 1 + 0.4 \times 2 = 1.4$ days. The total cycle time is then $1 + 3 + 3 + 1.4 = 8.4$ days.

In this example the cycle time is in great part determined by task "Check income sources", which is the one that determines the cycle time of the fragment between the AND-split and the AND-join. In this case, we say that this task is part of the *critical path* of the process. The critical path of a process is the sequence of tasks that determines the cycle time of the process, meaning that the cycle time of any instance of the process is never lower than the sum of the cycle times of this sequence of

Table 7.1 Cycle times for credit application process

Activity	Cycle time
Check completeness	1 day
Check credit history	1 day
Check income sources	3 days
Assess application	3 days
Make credit offer	1 day
Notify rejection	2 days

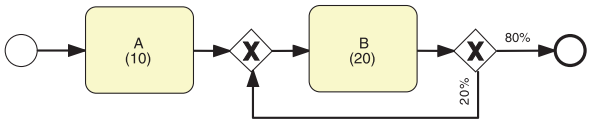


Fig. 7.7 Example of a rework loop

tasks. When optimizing a process with respect to cycle time, one should focus the attention on tasks that belong to the critical path.

Exercise 7.2 Consider the process model given in Fig. 3.8 (p. 73). Calculate the cycle time under the following assumptions:

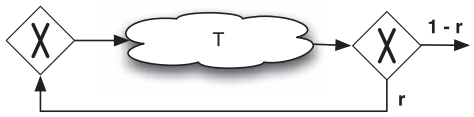
- Each task in the process takes 1 hour on average.
- In 40 % of the cases the order contains only Amsterdam products.
- In 40 % of the cases it contains only Hamburg products.
- In 20 % of the cases it contains products from both warehouses.

Compare the process model in Fig. 3.8 (p. 73) with the one in Fig. 3.10 (p. 74). Does this comparison give you an idea of how to calculate cycle times for process models with OR gateways?

Another recurrent case worth considering is the case where a fragment of a process may be repeated multiple times. This situation is called *rework* and is illustrated in Fig. 7.7. Here the decimal numbers attached to the arcs denote the probability that the corresponding arc will be taken. For sure, we can say that activity B will be executed once. Next, we can say that activity B may be executed twice with a probability of 20 % (i.e. 0.2), which is the probability of going back from the XOR-split gateway to the XOR-join gateway. If we continue this reasoning, we find out that the probability that task B is executed three times is $0.2 \times 0.2 = 0.04$, and more generally, the probability that task B is executed N times is 0.2^N.

If we sum up the cycle times in the cases where B is executed once, twice, three times, etc., we get the following summation $\sum_{i=0}^{\infty} 0.2^i$. In essence, this is the number of times that B is expected to be executed. If we replace 0.2 with a variable r, this summation is a well-known series, known as the *geometric series* and it can be shown that this series is equivalent to $1/(1-r)$. Hence, the average number of times that B is expected to be executed is $1/(1-0.2) = 1.25$. Now, if we multiply

Fig. 7.8 Rework pattern



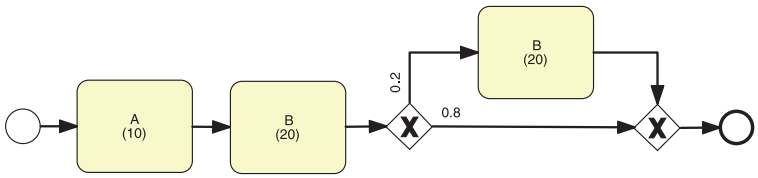


Fig. 7.9 Activity that is reworked at most once

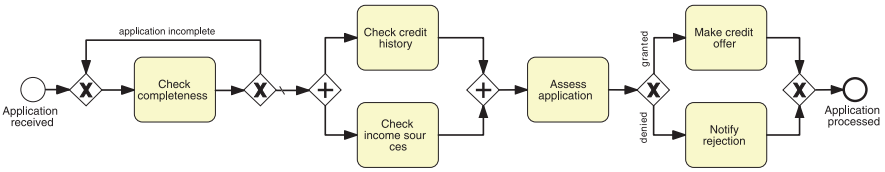


Fig. 7.10 Credit application process with rework

this expected number of instances of B times the cycle time of activity B, we get $1.25 \times 20 = 25$. Thus the total cycle time of the process in Fig. 7.7 is $10 + 25 = 35$.

More generally, the cycle time of the fragment with the structure shown in Fig. 7.8 is

$$CT = \frac{T}{1 - r}. \tag{7.3}$$

In this formula, parameter r is called the *rework probability*, that is, the probability that the fragment inside the cycle will need to be reworked. From here on, this type of block will be called a *rework block*, or more generally a *repetition block*.

In some scenarios, an activity is reworked at most once. This situation would be modeled as shown in Fig. 7.9. Using what we have seen, we can already calculate the cycle time of this example. First, we observe that the cycle time of the fragment between the XOR-split and the XOR-join is $0.2 \times 20 + 0.8 \times 0 = 4$. Here, the zero comes from the fact that one of the branches between the XOR-split and the XOR-join is empty and therefore does not contribute to the cycle time. To complete this, we have to add the cycle time of the preceding activities, giving us a total cycle time of 34.

Example 7.3 Let us consider the credit application process model in Fig. 7.10 and the cycle times previously given in Table 7.1. Let us also assume that in 20 % of the cases, the application is incomplete and in 60 % of cases the credit is granted.

The cycle time of the rework block is $10/(1 - 0.2) = 1.25$ days. The cycle time of the AND-block is 3 days and that of the XOR-block is 1.4 days as discussed in Example 7.2. Thus the total cycle time is $1.25 + 3 + 3 + 1.4 = 8.65$ days.

7.2.2 Cycle Time Efficiency

As previously mentioned, the cycle time of an activity or of a process can be divided into *waiting time* and *processing time*. Waiting time is the portion of the cycle time where no work is being done to advance the process. This includes time spent in transferring information about the case between process participants, like for example when documents are exchanged by post, as well as time when the case is waiting for an actor to process it. Processing time on the other hand refers to the time that actors spend doing actual work. In many if not most processes, the waiting time makes up a considerable proportion of the overall cycle time. There are a variety of reasons for this phenomenon. For example, this situation may happen because work is performed in batches. In a process related to the approval of purchase requisitions at a company, the supervisor responsible for such approvals in a business unit might choose to batch all applications and check them only once at the start or the end of a working day. Also, sometimes waiting time is spent waiting for an external actor to provide some input for a task. For example, in the context of fulfilling a medical prescription, a pharmacist may require a clarification from the doctor. To do so, the pharmacist would try to call the doctor. But the doctor might be unavailable and so the pharmacist needs to put the prescription aside and wait until the doctor returns the call.

When analyzing a process with the aim of addressing issues related to cycle time, it may be useful to start by evaluating the ratio of overall processing time relative to the overall cycle time. This ratio is called *cycle time efficiency*. A cycle time efficiency close to 1 indicates that there is little room for improving the cycle time unless relatively radical changes are introduced in the process. A ratio close to zero indicates that there is a significant amount of room for improving cycle time by reducing the waiting time, for example due to handovers between participants.

The cycle time efficiency of a process can be calculated as follows. First, we need to determine the cycle time and the processing time of each activity. Given this information, we can then calculate the overall cycle time of the process using the same formulas we saw above. Let us call this amount CT. Next, using the same formulas, we can calculate the overall amount of time that is spent doing actual work. This is called the *theoretical cycle time* of the process. Essentially, this is the amount of time that an instance of the process would take if there was no waiting time at all. To calculate the theoretical cycle time, we apply the same method as for calculating cycle time, but instead of using the cycle time of each activity, we use the processing time of each activity. Let us call the theoretical cycle time TCT. The cycle time efficiency (CTE) is then calculated as follows:

$$CTE = \frac{TCT}{CT}$$

Example 7.4 Let us consider the credit application process model in Fig. 7.10 and the processing times given in Table 7.2. The activity cycle times (including both waiting and processing time) are those previously given in Table 7.1. Let us assume

Table 7.2 Processing times for credit application process

Activity	Cycle time
Check completeness	2 hours
Check credit history	30 minutes
Check income sources	3 hours
Assess application	2 hours
Make credit offer	2 hours
Notify rejection	30 minutes

Table 7.3 Activity cycle times and processing times for ministerial enquiry process

Activity	Cycle time	Processing time
Register ministerial enquiry	2 days	30 mins
Investigate ministerial enquiry	8 days	12 hours
Prepare ministerial response	4 days	4 hours
Review ministerial response	4 days	2 hour

that in 20 % of cases, the application is incomplete and in 60 % of cases the credit is "granted". Let us additionally assume that one day is equal to 8 working hours.

We have seen in Example 7.3 that the total cycle time of this process is 8.65 days, which translates to 69.2 working hours. We now calculate the theoretical cycle time in the same way as the total cycle time but using the processing times given in Table 7.2. This gives us: $2/(1-0.2)+3+2+0.6\times 2+0.4\times 0.5=9.9$ working hours. The cycle time efficiency is thus $8.9/69.2=12.9$ %.

Exercise 7.3 Calculate the overall cycle time, theoretical cycle time and cycle time efficiency of the ministerial enquiry process introduced in Exercise 3.7 (p. 77). Assume that the rework probability is 0.2 and that the waiting times and processing times are those given in Table 7.3.

7.2.3 Cycle Time and Work-In-Process

Cycle time is directly related to two measures that play an important role when analyzing a process, namely *arrival rate* and *Work-In-Process* (WIP).

The arrival rate of a process is the average number of new instances of the process that are created per time unit. For example, in a credit application process, the arrival rate is the number of credit applications received per day (or any other time unit we choose). Similarly, in an order-to-cash process, the arrival rate is the average number of new orders that arrive per day. Traditionally, the symbol λ is used to refer to the arrival rate.

Meanwhile, WIP is the average number of instances of a process that are active at a given point in time, meaning the average number of instances that have not

yet completed. For example, in a credit application process, the WIP is the average number of credit applications that have been submitted and not yet granted or rejected. Similarly, in an order-to-cash process, the WIP is the average number of orders that have been received but not yet delivered and paid.

Cycle time (CT), arrival rate (λ) and WIP are related by a fundamental law known as Little's law, which states that:

$$WIP = \lambda \times CT$$

Basically what this law tells us is that:

- WIP increases if the cycle time increases or if the arrival rate increases. In other words, if the process slows down—meaning that its cycle time increases—there will be more instances of the process active concurrently. Also, the faster new instances are created, the higher will be the number of instances in an active state.
- If the arrival rate increases and we want to keep the WIP at current levels, the cycle time must decrease.

Little's law holds for any stable process. By stable, we mean that the number of active instances is not increasing infinitely. In other words, in a stable process, the amount of work waiting to be performed is not growing beyond control.

Although simple, Little's law can be an interesting tool for what-if analysis. We can also use Little's law as an alternative way of calculating total cycle time of a process if we know the arrival rate and WIP. This can be useful sometimes because determining the arrival rate and WIP is sometimes easier than determining the cycle time. For example, in the case of the credit application process, the arrival rate can be easily calculated if we know the total number of applications processed over a period of time. For example, if we assume there are 250 business days per year and we know the total number of credit applications over the last year is 2500, we can infer that the average number of applications per business day is 10. WIP on the other hand can be calculated by means of sampling. We can ask how many applications are active at a given point in time, then ask this question again one week later and again two weeks later. Let us assume that on average we observe that 200 applications are active concurrently. The cycle time is then $WIP/\lambda = 200/10 = 20$ business days.

Exercise 7.4 A restaurant receives on average 1,200 customers per day (between 10am and 10pm). During peak times (12pm to 3pm and 6pm to 9pm), the restaurant receives around 900 customers in total and, on average, 90 customers can be found in the restaurant at a given point in time. At non-peak times, the restaurant receives 300 customers in total and, on average, 30 customers can be found in the restaurant at a given point in time.

- What is the average time that a customer spends in the restaurant during peak times?
- What is the average time that a customer spends in the restaurant during non-peak times?

- The restaurant's premises have a maximum capacity of 110 customers. This maximum capacity is sometimes reached during peak times. The restaurant manager expects that the number of customers during peak times will increase slightly in the coming months. What can the restaurant do to address this issue without investing in extending its building?

7.2.4 Other Applications and Limitations of Flow Analysis

As mentioned earlier, flow analysis can also be used to calculate other performance measures besides cycle time. For example, assuming we know the average cost of each activity, we can calculate the cost of a process more or less in the same way as we calculate cycle time. In particular, the cost of a sequence of activities is the sum of the costs of these activities. Similarly the cost of an XOR-block is the weighted average of the cost of the branches of the XOR-block, and the cost of a rework pattern such as the one shown in Fig. 7.8 is the cost of the body of the loop divided by $1 - r$. The only difference between calculating cycle time and calculating cost relates to the treatment of AND-blocks. The cost of an AND-block such as the one shown in Fig. 7.5 is not the maximum of the cost of the branches of the AND-block. Instead, the cost of such a block is the sum of the costs of the branches. This is because after the AND-split is traversed, every branch in the AND join is executed and therefore the costs of these branches add up to one another.

Example 7.5 Let us consider again the credit application process model in Fig. 7.10 and the processing times given in Table 7.2. As previously, we assume that in 20 % of cases, the application is incomplete and in 60 % of cases the credit is granted. We further assume that activities "Check completeness", "Check credit history" and "Check income sources" are performed by a clerk, while activity "Assess application", "Make credit offer" and "Notify rejection" are performed by a credit officer. The hourly cost of a clerk is 25 while the hourly cost of a credit officer is 50. Performing a credit history requires that the bank submits a query to an external system. The bank is charged € 1 per query by the provider of this external system.

From this scenario, we can see that the cost of each task can be split into two components: the *human resource cost* and *other costs*. The human resource cost is the cost of the human resource(s) that performs the task. This can be calculated as the product of the hourly cost of the resource and the processing time (in hours) of the task. Other costs correspond to costs that are incurred by an execution of a task, but are not related to the time spent by human resources in the task. In this example, the cost per query to the external system would be classified as "other costs" for task "Check credit history". The remaining tasks do not have an "other cost" component. For the example at hand, the breakdown of resource cost, other cost and total cost per task is given in Table 7.4. Given this input, we can calculate the total cost-per-execution of the process as follows: $50/(1 - 0.2) + 13.5 + 75 + 100 + 0.6 \times 100 + 0.4 \times 25 = 321$.

Table 7.4 Cost calculation table for credit application process

Activity	Resource cost	Other cost	Total cost
Check completeness	$2 \times 25 = 50$	0	50
Check credit history	$0.5 \times 25 = 12.5$	1	13.5
Check income sources	$3 \times 25 = 75$	0	75
Assess application	$2 \times 50 = 100$	0	50
Make credit offer	$2 \times 50 = 100$	0	100
Notify rejection	$0.5 \times 50 = 25$	0	25

Exercise 7.5 Calculate the cost-per-execution of the ministerial enquiry process introduced in Exercise 3.7 (p. 77). Assume that the rework probability is 0.2 and that the times are those given in Table 7.3. Activity "Register ministerial enquiry" is performed by a clerk, activity "Investigate ministerial enquiry" is performed by an adviser, "Prepare ministerial response" is performed by a senior adviser, and "Review ministerial response" is performed by a minister counselor. The hourly resource cost of a clerk, adviser, senior adviser and minister counselor are 25, 50, 75, and 100, respectively. There are no other costs attached to these activities besides the resource costs.

Before closing the discussion on flow analysis, it is important to highlight some of its pitfalls and limitations. First of all, we should note that the equations presented in Sect. 7.2.1 do not allow us to calculate the cycle time of any process model. In fact, these equations only work in the case of block-structured process models. In particular, we cannot use these equations to calculate the cycle time of an unstructured process model such as the one shown in Exercise 3.9 (p. 93). Indeed, this example does not fit into any of the patterns we have seen above. Calculating the cycle time in this case is trickier. Also, if the model contains other modeling constructs besides AND and XOR gateways, the method for calculating cycle time becomes more complicated.

Fortunately, this is not a fundamental limitation of flow analysis, but only a limitation of the specific set of equations discussed in Sect. 7.2.1. There are other more sophisticated flow analysis techniques that allow us to calculate the cycle time of virtually any process model. The maths can get a bit more complex and in practice, one would not do such calculations by hand. But this is generally not a problem given that several modern process modeling tools include functionality for calculating cycle time, cost, and other performance measures of a process model using flow analysis.

A more fundamental roadblock faced by analysts when applying flow analysis is the fact that they first need to estimate the average cycle time of each activity in the process model. In fact, this obstacle is typical when applying any quantitative process analysis technique. There are at least two approaches to address this obstacle. The first one is based on interviews or observation. In this approach, analysts interview the stakeholders involved in each task or they observe how the stakeholders work during a given day or period of time. This allows analysts to at least make

an "informed guess" regarding the average time a case spends in each activity, both in terms of waiting time and processing time. A second approach is to collect logs from the information systems used in the process. For example, if a given activity such as approving a purchase requisition is performed by means of an internal Web portal (an Intranet), the administrators of the portal should be able to extract logs from this portal that would allow the analyst to estimate the average amount of time that a requisition form spends in "waiting for approval" mode and also the average time between the moment the supervisor opens a purchase requisition for approval and the time they approve it. With careful analysis, these logs can provide a wealth of information that can be combined via flow analysis to get an overall picture of which parts of the process consume the most time.

A major limitation of flow analysis is that it does not take into account the fact that a process behaves differently depending on the load, that is, depending on the amount of instances of the process that are running concurrently. Intuitively, the cycle time of a process for handling insurance claims would be much slower if the insurance company is handling thousands of claims at once, due for example to a recent natural disaster such as a storm, versus the case where the load is low and the insurance company is only handling a hundred claims at once. When the load goes up and the number of resources (e.g. claim handlers) remains relatively constant, it is clear that the waiting times are going to be longer. This is due to a phenomenon known as *resource contention*. Resource contention occurs when there is more work to be done than resources available to perform the work, like for example more claims than insurance claim handlers. In such scenarios, some tasks will be in waiting mode until one of the necessary resources are freed up. Flow analysis does not take into account the effects of increased resource contention. Instead, the estimates obtained from flow analysis are only applicable if the level of resource contention remains relatively stable over the long-run.

7.3 Queues

Queueing theory is a collection of mathematical techniques to analyze systems that have resource contention. Resource contention inevitably leads to queues as we all probably have experienced in supermarket check-out counters, at a bank's office, post office or government agency. Queueing theory gives us techniques to analyze important parameters of a queue such as the expected length of the queue or the expected waiting time of an individual case in a queue.

7.3.1 Basics of Queueing Theory

In basic queueing theory, a *queueing system* is seen as consisting of one or multiple *queues* and a *service* that is provided by one or multiple *servers*. The elements inside

a queue are called *jobs* or *customers*, depending on the specific context. For example, in the case of a supermarket, the service is that of checking out. This service is provided by multiple cashiers (the servers). Meanwhile, in the case of a bank office, the service is to perform a banking transaction, the servers are tellers, and there is generally a single queue that leads to multiple servers (the tellers). These two examples illustrate an important distinction between multi-line (i.e. multi-queue) queueing systems (like the supermarket) and single-line queueing systems (like the bank office).

Queueing theory provides a very broad set of techniques. It would be unrealistic to introduce all these techniques in this chapter. So instead of trying to present everything that queueing theory has to offer, we will present two queueing theory models that are relatively simple, yet useful when analyzing business processes or activities within a process.

In the two models we will be presenting, there is a single queue (single-line queueing system). Customers come at a given mean arrival rate that we will call λ. This is the same concept of arrival rate that we discussed above when presenting Little's law. For example, we can say that customers arrive at the bank office at a mean rate of 20 per hour. This implies that, on average, one customer arrives every 5 minutes ($\frac{1}{20}$ hour). This latter number is called the mean *inter-arrival time*. We observe that if λ is the arrival rate per time unit, then $1/\lambda$ is the mean inter-arrival time.

It would be illusory to think that the time between the arrival of two customers at the post office is always 5 minutes. This is just the mean value. In practice, customers arrive independently from one another, so the time between the arrival of one customer and the arrival of the next customer is completely random. Moreover, let us say that the time between the arrival of the first customer and the arrival of the second customer is 1 minute. This observation does not tell us absolutely anything about the time between the arrival of the second customer and the arrival of the third customer. It might be that the third customer arrives 1 minute after the second, or 5 minutes or 10 minutes. We will not know until the third customer arrives.

Such an arrival process is called a *Poisson process*. In this case, the distribution of arrivals between any two consecutive customers follows a so-called *exponential distribution* (specifically a *negative exponential distribution*) with a mean of $1/\lambda$. In a nutshell, this means that the probability that the inter-arrival time is exactly equal to t (where t is a positive number) decreases in an exponential manner when t increases. For instance, the probability of the time of inter-arrival time being 10 is considerably smaller than the probability of the inter-arrival time being 1. Hence, shorter inter-arrival times are much more probable than longer ones, but there is always a probability (perhaps a very small one) that the inter-arrival time will be a large number.

In practice, the Poisson process and the exponential distribution describe a large class of arrival processes that can be found in business processes, so we will be using them to capture the arrival of jobs or customers into a business process or an activity in a business process. The Poisson process can also be observed for example when

we examine how often cars enter a given segment of a highway, or how often calls go through a telephone exchange.

Having said this, one must always cross-check that cases arrive to a given process or activity in an exponentially distributed manner. This cross-check can be done by recording the inter-arrival times for a given period of time, and then feeding these numbers into a statistical tool such as for example R, Mathworks's Statistical Toolbox or EasyFit. These tools allow one to input a set of observed inter-arrival times and check if it follows a negative exponential distribution.

Exponential distributions are not only useful when modeling the inter-arrival time. They are also in some cases useful when describing the processing time of an activity.[1] In the case of activities that require a diagnosis, a non-trivial verification or some non-trivial decision making, it is often the case that the activity's processing time is exponentially distributed. Take for example the amount of time it takes for a mechanic to make a repair on a car. Most repairs are fairly standard, and the mechanics might take for example one hour to do them. However, some repairs are very complex, and in such cases, it can take the mechanic several hours to complete. A similar remark can be made of a doctor receiving patients in an emergency room. A large number of emergencies are quite standard and can be dispatched in less than an hour, but some emergencies are extremely complicated and can take hours to deal with. So it is likely that such activities will follow an exponential distribution. As mentioned above, when making such a hypothesis, it is important that you first check it by taking a random sample of processing times and feeding them to a statistical tool.

In the queueing theory field, a single-queue system is called an *M/M/1 queue* if the inter-arrival times of customers follow an exponential distribution, the processing times follow an exponential distribution, there is one single server and jobs are served on a First-In-First-Out (FIFO) basis. In the case of M/M/1 queue, we also assume that when a job arrives, it enters the queue and it stays there until it is taken on by the server.

If the above conditions are satisfied, but there are multiple servers instead of a single server, the queueing system is said to be *M/M/c*, where *c* is number of servers. For example, a queue is M/M/5 if the inter-arrival times of customers follow an exponential distribution, the processing times follow an exponential distribution and there are five servers at the end of the queue. The "M" in this denomination stands for "Markovian", which is the name given to the assumptions that inter-arrival times and processing times follow an exponential distribution. Other queueing models exist that make different assumptions. Each such model is different, so the results we will obtain for an M/M/1 or M/M/c queue are quite different from those we would obtain from other distributions.

[1] In queueing theory, the term service time is used instead of processing time. For uniformity purposes, here we use the term processing time.

7.3.2 M/M/1 and M/M/c Models

To summarize the previous discussion, an M/M/1 queue or M/M/c queue can be defined by means of the following parameters:

- λ—the mean arrival rate per time unit. The mean inter-arrival time is then $1/\lambda$. For example, $\lambda = 5$ means that there are 5 arrivals per hour and this entails that the mean inter-arrival time between two consecutive jobs is $1/5$ hours, that is 12 minutes.
- μ—the mean number of customers that can be served per time unit. The mean processing time per job is then $1/\mu$. For example, $\mu = 6$ means six jobs are served per hour, that is, one job is served in 10 minutes (on average).
- In the case of M/M/c, the number of servers (c).

Given parameters λ and μ, we can already state how busy a server is. This is called the occupation rate ρ and is equal to λ/μ. In the above example, the occupation rate is $5/6 = 83.34$ %. It should be noted that this is a relatively high occupation rate. A system with an occupation rate of more than 100 % is unstable, meaning that the queue will become longer and longer forever because the server cannot cope with all the demand. In fact, even a system at close to 100 % of occupation rate is unstable because of the randomness at which new jobs arrive and the variability in the processing times per job. To understand why this is the case, just imagine if you were a doctor receiving patients at a rate of 6 per hour for 8 hours, knowing that every patient takes 10 minutes on average to be treated (sometimes less but sometimes more). Without any slack, most likely you will end up with a tremendous backlog at the end of the day.

In the case of an M/M/c system, the occupancy rate is $\frac{\lambda}{c\mu}$ since the system can handle jobs at a rate of $c\mu$. For example, if the system has two servers and each server can handle two jobs per hour, the system can handle four jobs per hour. If jobs arrive at a mean rate of 3 per hour, the occupancy rate of the system is $3/4 = 75$ %.

Given an M/M/1 or M/M/c system, queueing theory allows us to calculate the following parameters:

- L_q—The average number of jobs (e.g. customers) in the queue.
- W_q—The average time one job spends in the queue.
- W—The average time one job spends in the system. This includes both the time the customer spends in the queue but also the time the customer spends being serviced.
- L—The average number of jobs in the system (i.e. the Work-in-Progress referenced in Little's law).

To summarize, the general structure of a single-queue system—consisting of one queue and one or many servers—is depicted in Fig. 7.11. The parameters of the queue (λ, c and μ) are shown at the top. The parameters that can be computed from these three input parameters are shown under the queue and the server. The average time a job waits in the queue is W_q, while the average length of the queue is L_q. Eventually, a job goes into the server and in there it spends on average $1/\mu$

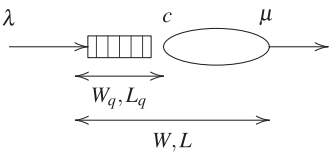


Fig. 7.11 Structure of an M/M/1 or M/M/c system, input parameters and computable parameters

time units. The average time between the moment a job enters the system and the moment it exits is W, while the average number of jobs inside the system (in the queue or in a server) is L.

Queueing theory gives us the following formulas for calculating the above parameters for M/M/1 models:

$$L_q = \rho^2/(1-\rho) \tag{7.4}$$

$$W_q = \frac{L_q}{\lambda} \tag{7.5}$$

$$W = W_q + \frac{1}{\mu} \tag{7.6}$$

$$L = \lambda W \tag{7.7}$$

Formulas (7.5), (7.6), and (7.7) can be applied to M/M/c models as well. The only parameter that needs to be calculated differently in the case of M/M/c models is L_q. For M/M/c models, L_q is given by the following formula:

$$L_q = \frac{(\lambda/\mu)^c \rho}{c!(1-\rho)^2(\frac{(\lambda/\mu)^c}{c!(1-\rho)} + \sum_{n=0}^{c-1} \frac{(\lambda/\mu)^n}{n!})} \tag{7.8}$$

The formula for computing L_q in the case of M/M/c models is particularly complicated because of the summations and factorials. Fortunately, there are tools that can do this for us. For example, the Queueing Toolpack[2] supports calculations for M/M/c systems (called *M/M/s* in the Queueing Toolpack) as well as M/M/c/k systems, where k is the maximum number of jobs allowed in the queue. Jobs that arrive when the length of the queue is k are rejected (and may come back later). Other tools for analyzing queueing systems include QSim[3] and PDQ.[4]

Example 7.6 A company designs customized electronic hardware for a range of customers in the high-tech electronics industry. The company receives orders for designing a new circuit every 20 working days on average. It takes a team of engineers on average 10 working days to design a hardware.

This problem can be mapped to an M/M/1 model assuming that the arrival of designs follows a Poisson process, that the distribution of times for designing a circuit follows an exponential distribution and that new design requests are handled

[2] http://apps.business.ualberta.ca/aingolfsson/qtp/.

[3] http://www.stat.auckland.ac.nz/~stats255/qsim/qsim.html.

[4] http://www.perfdynamics.com/Tools/PDQ.html.

on a FIFO manner. Note that even though the team includes several people, they act as a monolithic entity and therefore it should be treated as a single server.

We will hereby take the working day as a time unit. On average, 0.05 orders are received per day ($\lambda = 0.05$), and 0.2 orders are fulfilled per day ($\mu = 0.1$). Thus, the occupation rate of this system $\rho = 0.05/0.1 = 0.5$. Using the formulas for M/M/1 models, we can deduce that the average length of the queue L_q is: $0.5^2/(1 - 0.5) = 0.5$ orders. From there we can conclude that the average time an order spends on the queue is $W_q = 0.5/0.05 = 10$ days. Thus, it takes on average order $W = 10 + 1/0.1 = 20$ working days for an order to be fulfilled.

Exercise 7.6 Consider now the case where the engineering team in the previous example takes 16 working days to design a hardware. What is then the average amount of time an order takes to be fulfilled?

Exercise 7.7 An insurance company receives 220 calls per day from customers who want to lodge an insurance claim. The call center is open from 8am to 5pm. The arrival of calls follows a Poisson process. Looking at the intensity of arrival of calls, we can distinguish three periods during the day: the period 8am to 11am, the period 11am to 2pm and the period 2pm to 5pm. During the first period, around 60 calls are received. During the 11am–2pm period, 120 calls are received, and during the 2pm–5pm period, 40 calls are received. A customer survey has shown that customers tend to call between 11am and 2pm because during this time they have a break at work and they take advantage of their break to make their personal calls.

Statistical analysis shows that the durations of calls follow an exponential distribution.

According to the company's customer service charter, customers should wait no more than one minute on average for their call to be answered.

- Assume that the call center can handle 70 calls per hour using seven call center agents. Is this enough to meet the 1-minute constraint set in the customer service charter? Please explain your answer by showing how you calculate the average length of the queue and the average waiting time.
- What happens if the call center's capacity is increased so that it can handle 80 calls per hour (using eight call center agents)?
- The call center manager has been given a mandate to cut costs by at least 20 %. Give at least two ideas to achieve this cut without reducing the salaries of the call center agents and while keeping an average waiting time below or close to one minute.

7.3.3 Limitations of Basic Queueing Theory

The basic queueing analysis techniques presented above allow us to estimate waiting times and queue length based on the assumptions that inter-arrival times and processing times follow an exponential distribution. When these parameters follow

different distributions, one needs to use very different queueing models. Fortunately, queueing theory tools nowadays support a broad range of queueing models and of course they can do the calculations for us. The discussion above was intended as an overview of single-queue models, with the aim of providing a starting point from where you can learn more about this family of techniques.

A more fundamental limitation of the techniques introduced in this section is that they only deal with one activity at a time. When we have to analyze an entire process that involves several activities, events, and resources, these basic techniques are not sufficient. There are many other queueing analysis techniques that could be used for this purpose, like for example queueing networks. Essentially, queueing networks are systems consisting of multiple inter-connected queues. However, the maths behind queueing networks can become quite complex, especially when the process includes concurrent activities. A more popular approach for quantitative analysis of process models under varying levels of resource contention is process simulation, as discussed below.

7.4 Simulation

Process simulation is arguably the most popular and most widely supported technique for quantitative analysis of process models. The basic idea underpinning process simulation is quite simple. In essence, a process simulator generates a large number of hypothetical instances of a process, executes these instances step-by-step, and records each step in this execution. The output of a simulator typically includes the logs of the simulation as well as some statistics related to cycle times, average waiting times and average resource utilization.

7.4.1 Anatomy of a Process Simulation

During a process simulation, the tasks in the process are not actually executed. Instead, the simulation of a task proceeds as follows. When a task is ready to be executed, a so-called *work item* is created and the simulator first tries to find a resource to which it can assign this work item. If no resource able to perform the work item is found, the simulator puts the work item in waiting mode until a suitable resource is freed up. Once a resource is assigned to a work item, the simulator determines the duration of the work item by drawing a random number according to the probability distribution of the task's processing time. This probability distribution and the corresponding parameters need to be defined in the simulation model.

Once the simulator has determined the duration of a work item, it puts the work item in sleeping mode for that duration. This sleeping mode simulates the fact that the task is being executed. Once the time interval has passed (according to the simulation's clock), the work item is declared to be completed, and the resource that was assigned to it becomes available.

In reality, the simulator does not effectively wait for tasks to come back from their sleeping mode. For example, if the simulator determines that the duration of a work item is 2 days and 2 hours, it will not wait for this amount of time to pass by. You can imagine how long a simulation would take if that was the case. Luckily, simulators use smart algorithms to complete the simulation as fast as possible. Modern business process simulators can effectively simulate thousands of process instances and tens of thousands of work items in a matter of seconds.

For each work item created during a simulation, the simulator records the identifier of the resource that was assigned to this instance as well as three time stamps:

- The time when the task was ready to be executed.
- The time when the task was started, meaning that it was assigned to a resource.
- The time when the task completed.

Using the collected data, the simulator can compute the average waiting time for each task. These measures are quite important when we try to identify bottlenecks in the process. Indeed, if a task has a very high average waiting time, it means that there is a bottleneck at the level of this task. The analyst can then consider several options for addressing this bottleneck.

Additionally, since the simulator records which resources perform which work items and it knows how long each work item takes, the simulator can find out the total amount of time during which a given resource is busy handling work items. By dividing the amount of time that a resource was busy during a simulation by the total duration of the simulation, we obtain the *resource utilization*, that is, the percentage of time that the resource is busy on average.[5]

7.4.2 Input for Process Simulation

From the above description of how a simulation works, we can see that the following information needs to be specified for each task in the process model in order to simulate it:

- Probability distribution for the processing time of each task.
- Other performance attributes for the task such as cost and added-value produced by the task.
- The set of resources that are able to perform the task. This set is usually called a *resource pool*. For example, a possible resource pool could be the "Claim Handlers" or "Clerks" or "Managers". Separately, the analyst needs to specify for each resource pool the number of resources in this pool (e.g. the number of claim handlers or the number of clerks) and other attributes of these resources such as the hourly cost (e.g. the hourly cost of a claims handler).

[5]Note that when discussing queueing theory above, we used the term occupation rate instead of resource utilization. These two terms are synonyms.

Common probability distributions for task durations in the context of process simulation include:

- *Fixed.* This is the case where the processing time of the task is the same for all executions of this task. It is rare to find such tasks because most tasks, especially those involving human resources, would exhibit some variability in their processing time. Examples of tasks with fixed processing time can be found among automated tasks such as for example a task that generates a report from a database. Such a task would take a relatively constant amount of time, say for example 5 seconds.
- *Exponential distribution*. As discussed in Sect. 7.3, the exponential distribution may be applicable when the processing time of the task is most often around a given mean value, but sometimes it is considerably longer. For example, consider a task "Assess insurance claims" in an insurance claims handling process. You can imagine that in most cases, the insurance claims fall within very standard cases. In such cases, the claim is assessed in an hour, or perhaps less. However, some insurance claims require special treatment, for example because the assessor considers that there is a risk that the claim is fraudulent. In this case, the assessor might spend several hours or even an entire day assessing a single claim. A similar observation can be made of diagnostics tasks, such as diagnosing a problem in an IT infrastructure, or diagnosing a problem during a car repair process.
- *Normal distribution*. This distribution is used when the processing time of the task is around a given average, and the "deviation" around this value is symmetric, meaning that the actual processing time can be above or below the mean with the same probability. Simple checks, such as for example checking whether or not a paper form has been fully completed might follow this distribution. Indeed, it generally takes about 3 minutes to make such a check. In some cases, this time can be lower because for example the form is clearly incomplete or clearly complete, and in other cases it can take a bit longer because a couple of fields have been left empty and it is unclear if these fields are relevant or not for the specific customer who submitted the form.

When assigning an exponential distribution to a task duration, the analyst has to specify the mean value. Meanwhile, when assigning a normal distribution, the analyst has to specify two parameters: mean value and standard deviation. These values are derived through an informed guess (based on interviews with the relevant stakeholders), but preferably by means of sampling (the analyst collects data for a sample of tasks executions) or by analyzing logs of relevant information systems. Some simulation tools allow the analyst to import logs into the simulation tool and assist the analyst in selecting the right probability distribution for task durations based on these logs. This functionality is called *simulation input analysis*.

In addition to the above per-task simulation data, a branching probability needs to be specified for every arc stemming from a decision gateway. These probabilities are determined by interviewing relevant stakeholders, observing executions of the process during a certain period of time, or collecting logs from relevant information systems.

Finally, in order to run a simulation, the analyst additionally needs to specify at least the following:

- The inter-arrival times and the mean arrival rate. As explained above, a very frequent distribution of inter-arrival times is the exponential distribution and this is usually the default distribution supported by business process simulators. It may happen, however, that the inter-arrival times follow a different distribution such as for example a *normal distribution*. By feeding a sample of inter-arrival times during a certain period of time to a statistical tool, we can find out which distribution best matches the data. Some simulators provide a module for selecting a distribution for the inter-arrival times and for computing the mean inter-arrival time from a data sample.
- The starting date and time of the simulation (e.g. "11 Nov. 2012 at 8:00").
- One of the following:
  - The end date and time of the simulation. If this option is taken, the simulation will stop producing more process instances once the simulation clock reaches the end time.
  - The real-time duration of the simulation (e.g. 7 days, 14 days). In this way, the end time of the simulation can be derived by adding this duration to the starting time.
  - The required number of process instances to be simulated (e.g. 1,000). If this option is taken, the simulator generates process instances according to the arrival rate until it reaches the required number of process instances. At this point, the simulation stops. Some simulators will not stop immediately, but will allow the active process instances to complete before stopping the simulation.

Example 7.7 We consider the process for loan application approval modeled in Fig. 4.6 (p. 104). We simulate this model using the BIMP simulator available at: http://bimp.cs.ut.ee. This simulator takes as input BPMN process models in XML format produced by other process modeling tools such as Signavio Process Editor or OpenText Provision. We provide the following inputs for the simulation.

- Two loan applications per hour, meaning an inter-arrival time of 30.
- Tasks "Check credit history" and "Check income sources" are performed by clerks.
- Tasks "Notify rejection", "Make credit offer" and "Assess application" are performed by credit officers.
- Task "Receive customer feedback" is in fact an event. It takes zero time and it only involves the credit information system (no human actors involved). To capture this, the task is assigned to a special "System" role.
- There are three clerks and three credit officers. The hourly cost of a clerk is € 25 while that of a credit officer is € 50.
- Clerks and credit officers work from 9am to 5pm during weekdays.
- The cycle time of task "Assess application" follows an exponential distribution with a mean of 20 minutes.

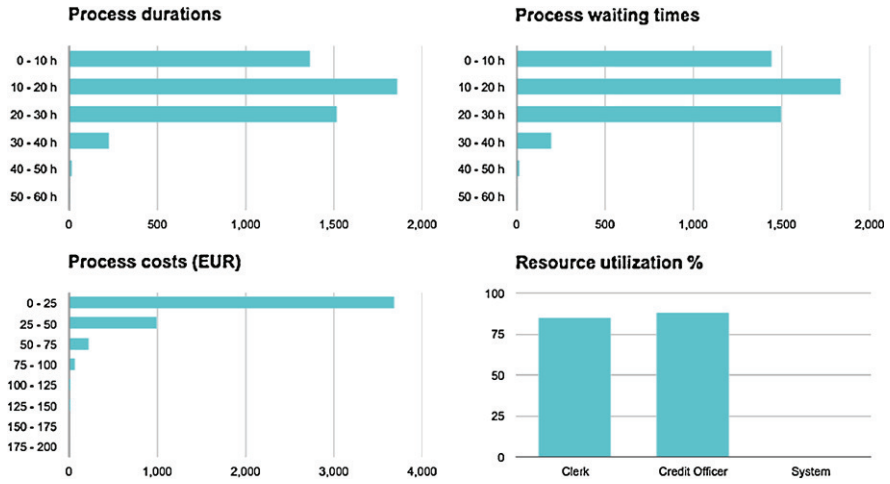


Fig. 7.12 Histograms produced by simulation of the credit application process

- Cycle times of all other tasks follow a normal distribution. Tasks "Check credit history", "Notify rejection" and "Make credit offer" have a mean cycle time of 10 minutes with a 20 % standard deviation, while "Check income sources" has a cycle time of 20 minutes with a 20 % standard deviation as well.
- The probability that an application is accepted is 80 %.
- The probability that a customer whose application was rejected, asks that the application be re-assessed is 20 %.

We run a simulation with 5,000 instances, which means around 104 days of loan applications arrivals assuming that applications arrive 24 hours a day, 7 days a week.[6] The simulation gives an average cycle time of around 17 hours. A variance of ± 2 hours can be observed when running the simulation multiple times. This variance is expected due to the stochastic nature of the simulation. Accordingly, it is recommended to run the simulation multiple times and to take averages of the simulation results. Figure 7.12 shows the histograms for process cycle time (called process duration in BIMP), waiting time (time a case spends waiting for resources to become available), cost (of resources), and resource utilization. It can be seen that applications spend most of the time in waiting mode, waiting for resources to become available. Resource utilization of clerks and credit officers is at 85 % and 88.5 %, respectively, meaning that there is some overload. As a rule of thumb, a resource utilization above 80 % means that one can expect long queues and high waiting times. If we add two clerks and two credit officers to the simulation we obtain an average cycle time of around 8 hours (compared to 17 hours) and an utilization rate of around 80 % for clerks and 50 % for credit officers.

[6] Some simulators additionally allow one to specify that new cases are only created during certain times of the day and certain days of the week, or according to a given calendar.

Exercise 7.8 An insurance company, namely Cetera, is facing the following problem: Whenever there is a major event (e.g. a storm), their claim-to-resolution process is unable to cope with the ensuing spike in demand. During normal times, the insurance company receives about 9,000 calls per week, but during a storm scenario, the number of calls per week doubles.

The claim-to-resolution process model of Cetera is presented in Fig. 7.13. The process starts when a call related to lodging a claim is received. The call is routed to one of two call centers depending on the location of the caller. Each call center receives approximately the same amount of calls (50–50) and has the same number of operators (40 per call center). The process for handling calls is identical across both call centers. When a call is received at a call center, the call is picked up by a call center operator. The call center operator starts by asking a standard set of questions to the customer to determine if the customer has the minimum information required to lodge a claim (e.g. insurance policy number). If the customer has enough information, the operator then goes through a questionnaire with the customer, enters all relevant details, checks the completeness of the claim and registers the claim.

Once a claim has been registered, it is routed by the claims handling office, where all remaining steps are performed. There is one single claims handling office, so regardless of the call center agent where the claim is registered, the claim is routed to the same office. In this office, the claim goes through a two-stage evaluation process. First of all, the liability of the customer is determined. Secondly, the claim is assessed in order to determine if the insurance company has to cover this liability and to what extent. If the claim is accepted, payment is initiated and the customer is advised of the amount to be paid. The activities of the claims handling department are performed by *claims handlers*. There are 150 claims handlers in total.

The mean cycle time of each task (in seconds) is indicated in Fig. 7.13. For every task, the cycle time follows an exponential distribution. The hourly cost of a call center agent is 30, while hourly cost of a claims handler is 50.

Describe the input that should be given to a simulator in order to simulate this process in the normal scenario and in the storm scenario. Using a simulation tool, encode the normal and the storm scenarios and run a simulation in order to compare these two scenarios.

7.4.3 Simulation Tools

Nowadays, most business process modeling tools provide simulation capabilities. Examples of tools with simulation support include: ADONIS, ARIS Business Designer, IBM Websphere Business Modeler, OpenText ProVision, Oracle Business Process Analysis (BPA) Suite, Savvion Process Modeler, Signavio Process Editor and TIBCO Business Studio. The landscape of tools evolves continuously, and thus it is very useful to understand the fundamental concepts of process simulation before trying to grasp the specific features of a given tool.

In general, the provided functionality varies visibly from one tool to another. For example, some tools allow one to capture the fact that resources do not work

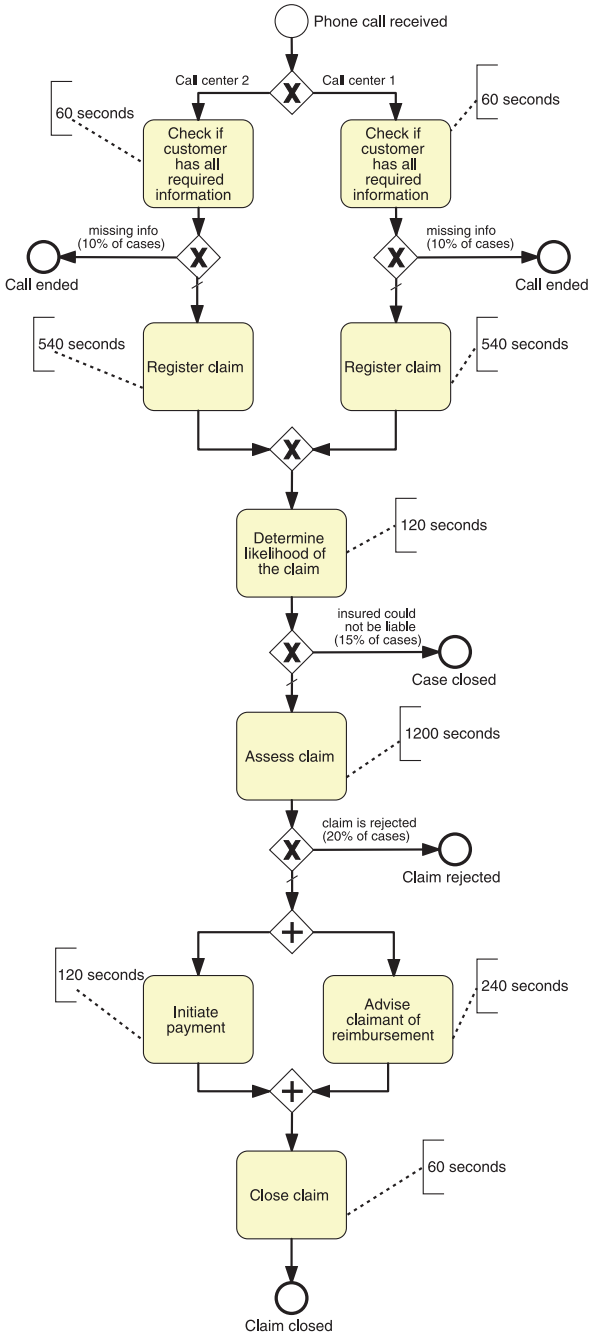


Fig. 7.13 Cetera's claim-to-resolution process

continuously, but only during specific periods of time. This is specified by attaching a calendar to each resource pool. Some tools additionally allow one to specify that new process instances are created only during certain periods of time, for example only during business hours. Again, this is specified by attaching a calendar to the process model.

Some of the more sophisticated tools allow one to specify not only branching conditions, but also actual boolean expressions that make use of attributes attached to data objects in the process model. In this way, we can specify for example that a branch coming out of an XOR-split should be taken when the attribute "loanAmount" of a data object called "loan application" is greater than 10,000, whereas another branch should be taken when this amount is lower than 10,000. In this case, the probabilistic distribution of values for the attribute "loanAmount" needs to be specified. When the simulator generates objects of type loan, it will give them a value according to the probability distribution attached to that attribute.

There are also small nuances between tools. For example some tools require one to specify the mean arrival rate, that is the number of cases that start during one time unit (e.g. 50 cases per day), while other tools require one to specify the mean inter-arrival time between cases (e.g. one case every 2 minutes). Recall that the distinction between mean arrival rate (written λ in queueing theory) and mean inter-arrival time ($1/\lambda$) was discussed in Sect. 7.3.1. Other tools go further by allowing one to specify not only the inter-arrival time, but how many cases are created every time. By default, cases arrive one by one, but in some business processes, cases may arrive in batches as illustrated by the following scenario extracted from a description of an archival process at the Macau Historical Archives:

> At the beginning of each year, transfer lists are sent to the Historical Archives by various organizations. Each transfer list contains approximately 225 historical records. On average two transfer lists are received each year. Each record in a transfer list needs to go through a process that includes appraisal, classification, annotation, backup, and re-binding among other tasks.
> If we consider that each record is a case of this archival process, then we can say that cases arrive in batches of $225 \times 2 = 450$ cases. Moreover, these batches arrive at a fixed inter-arrival time of one year.

Finally, process simulation tools typically differ in terms of how resource pools and resource costs are specified. Some tools would only allow one to define a resource pool and define the number of resources in the pool. A single cost per time unit is then attached to the entire resource pool. Other tools would allow one to create the resources of a pool one by one and to assign a cost to each created resource (e.g. create 10 clerks one by one, each with its name and hourly cost).

The above discussion illustrates some of the nuances found across simulation tools. In order to avoid diving straight away into the numerous details of a tool, it may be useful for beginners to take their first steps using the BIMP simulator referred to in Example 7.7. BIMP is a rather simple BPMN process model simulator that provides the core functionality found in commercial business process simulation tools.

7.4.4 A Word of Caution

One should keep in mind that the quantitative analysis techniques we have seen in this chapter, and simulation in particular, are based on models and on simplifying assumptions. The reliability of the output produced by these techniques largely depends on the accuracy of the numbers that are given as input. Additionally, simulation assumes that process participants work continuously on the process being simulated. In practice though, process participants are not robots. They get distracted due to interruptions, they display varying performance depending on various factors, and they may adapt differently to new ways of working.

In this respect, it is good practice whenever possible to derive the input parameters of a simulation from actual observations, meaning from historical process execution data. This is possible when simulating an as-is process that is being executed in the company, but not necessarily when simulating a to-be process. In a similar spirit, it is recommended to cross-check simulation outputs against expert advice. This can be achieved by presenting the simulation results to process stakeholders (including process participants). The process stakeholders are usually able to provide feedback on the credibility of the resource utilization levels calculated via simulation and the actual manifestation of the bottlenecks shown in the simulation. For instance, if the simulation points to a bottleneck in a given task, while the stakeholders and participants perceive this task to be uncritical, there is a clear indication that incorrect assumptions have been made. Feedback from stakeholders and participants helps to reconfigure the parameters such that the results come closer to matching the actual behavior. In other words, process simulation is an iterative analysis technique with potentially multiple validation loops.

Finally, it is advisable to perform sensitivity analysis of the simulation. Concretely, this means observing how the output of the simulation changes when adding one resource to or removing one resource from a resource pool, or when changing the processing times by ± 10 % for example. If such small changes in the simulation input parameters significantly affect the conclusions drawn from the simulation outputs, one can put a question mark on these conclusions.

7.5 Recap

In this chapter we saw three quantitative process analysis techniques, namely flow analysis, queueing theory and simulation. These techniques allow us to derive process performance measures, such as cycle time or cost, and to understand how different activities and resource pools contribute to the overall performance of a process.

Flow analysis allows us to calculate performance measures from a process model and performance data pertaining to each activity in the model. However, flow analysis does not take into account the level of busyness of the resources involved in the process, i.e. their level of resource utilization. Yet, waiting times are highly dependent on resource utilization—the busier the resources are, the longer the waiting times.

Basic queueing theory models, such as the M/M/1 model, allow us to calculate waiting times for individual activities given data about the number of resources and their processing times. Other queueing theory models such as queueing networks allow us to perform fine-grained analysis at the level of entire processes. However, in practice it is convenient to use process simulation for fine-grained analysis. Process simulation allows us to derive process performance measures (e.g. cycle time or cost) given data about the activities (e.g. processing times) and data about the resources involved in the process. Process simulation is a versatile technique supported by a range of process modeling and analysis tools.

7.6 Solutions to Exercises

Solution 7.1

1. There are at least two business processes that need improvement: the quote-to-booking process—which starts from the moment a quote is received to the moment that a booking is made—and the process for modifying bookings.
2. The quote-to-book process needs to be improved with respect to cycle time, and with respect to error rate. The booking modification process needs improvement with respect to error rate.

Solution 7.2 First we observe that the cycle time of the AND-block is 1. Next, we calculate the cycle time of the XOR-block as follows: $0.4 \times 1 + 0.4 \times 1 + 0.2 \times 1$ hour. The total cycle time is thus: $1 + 1 + 1 = 3$ hours.

Solution 7.3 The cycle time of the process is $2 + 8 + \frac{4+4}{1-0.2} = 20$ days. Assuming 8 working hours per day, this translates to 160 working hours. The theoretical cycle time is $0.5 + 12 + \frac{4+2}{1-0.2} = 20$ hours. Hence, cycle time efficiency is 12.5 %.

Solution 7.4 Little's law tells us that: $CT = WIP/\lambda$. At peak time, there are 900 customers distributed across 6 hours, so the mean arrival rate $\lambda = 150$ customers per hour. On the other hand, $WIP = 90$ during peak time. Thus, $CT = 90/150 = 0.6$ hours (i.e. 36 minutes). During non-peak time, $\lambda = 300/6 = 50$ customer per hour while $WIP = 30$, thus $CT = 30/50 = 0.6$ hours (again 36 minutes). If the number of customers per hour during peak times is expected to go up but the WIP has to remain constant, we need to reduce the cycle time per customer. This may be achieved by shortening the serving time, the interval between the moment a customer enters the restaurant and the moment they place an order, or the time it takes for the customer to pay. In other words, the process for order taking and payment may need to be redesigned.

Solution 7.5 Given that there are no other costs, we calculate the cost of the process by aggregating the resource costs as follows: $0.5 \times 25 + 12 \times 50 + (4 \times 75 + 2 \times 100)/(1 - 0.2) = 1237.50$.