It is common enough to see a diagram that looks something like the following:

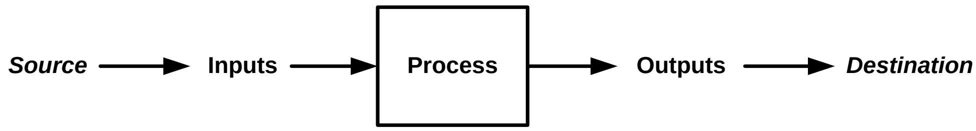

Figure 1

The diagram shows inputs coming from a source, passing through a process, and exiting as outputs headed to a destination. The box is labeled "process" but did you ever wonder what's inside the process box? The purposes of this paper are to discuss and explain what I think lies in there and to show how that view of what's inside the process box can be useful in figuring out how to improve the performance of people and processes.

Next, let's think for a moment about how inputs get converted into outputs. It's easy enough to say that a process converts or transforms inputs into outputs but just exactly how does it do that? Well, inside the process box is something known as "the processor" – the thingamajig that actually performs the operations that convert or transform the inputs into outputs. In some cases, the processor is a person; in others, it's a machine; and, in still others it's a combination of people and machines. (For example, consider me using my laptop and its software to produce this article.)

At this point, all we have inside the process box is a processor. However, there is also some activity going on in there: the processor is performing operations on the inputs. These operations – interactions between the processor and the inputs – result in the outputs. Moreover, these operations aren't performed in a willy-nilly way (at least we hope not). They are purposeful, controlled actions, intended to result in not just any old output but in a particular output that must meet some set of requirements. The processor, then, has two components: (1) the operator, which performs the operations and (2) the controller, which controls the operations and ensures that outputs satisfy requirements.

It's time for an aside. If the processor is a machine, we can, if we wish, separate the operator and the controller. We can also program the controller to make the operator do whatever we want. However, when the processor is a human being, the controller and the operator are a single, integrated unit. People control their own behavior. To get other people to perform the way we want entails recognizing and dealing with the fact that they are "living" control systems, pursuing their own goals and objectives and exercising control aimed at keeping things the way they want them. Getting people to do what we want is a far cry from making machines do what we want. But that's a different story for another day. As some writers say, "That's beyond the scope of this article." Now back to what's inside the process box.

Take a look at Figure 2 below and then we'll talk about it a bit.

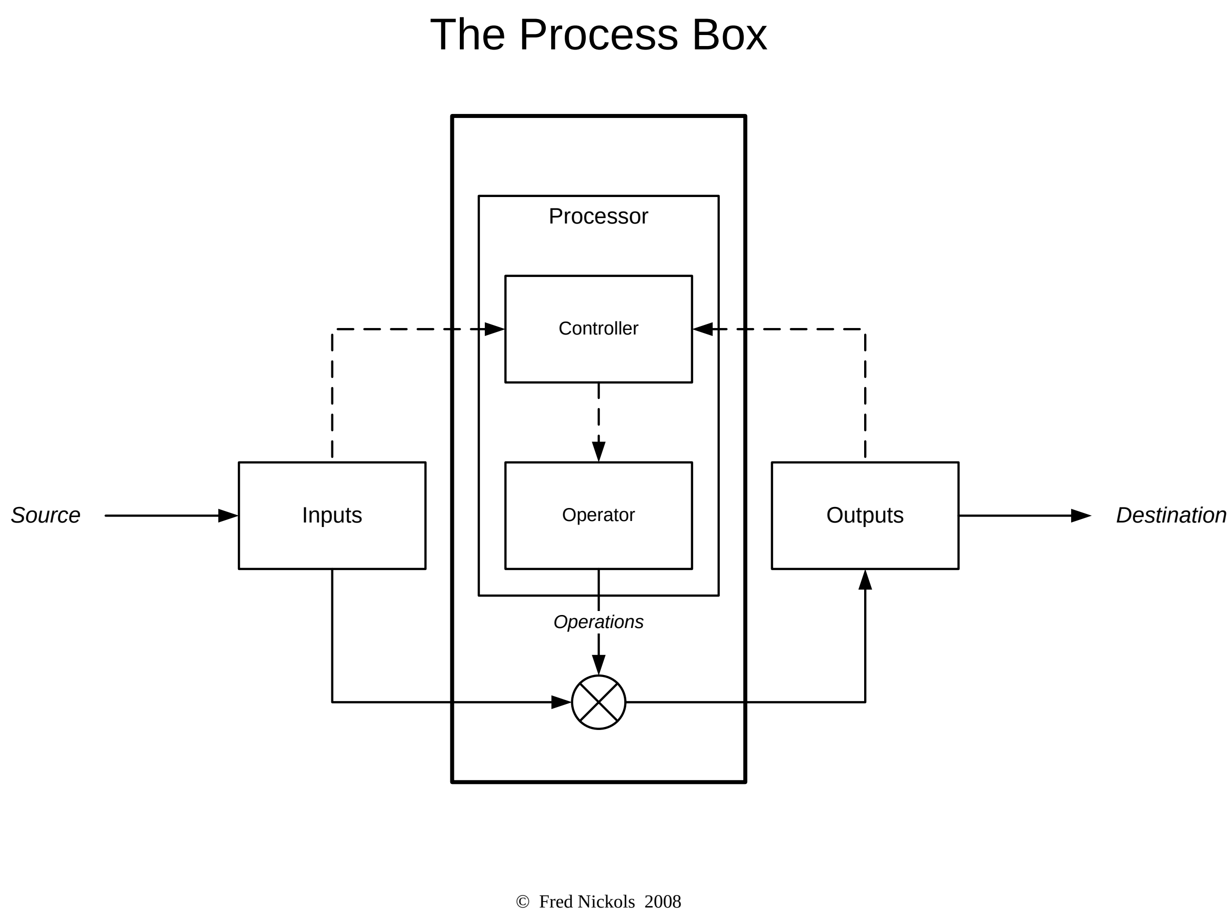

Figure 2

Our old friends – inputs and outputs – are there, as are their source and destination. So is the processor, including the controller and the operator. Also indicated are the operations performed by the operator. But what's that little circle with the X in it and what do the solid and dashed lines signify?

The circle with the X inside is an electro-mechanical computing symbol for an integrator-differentiator, a device capable of combining inputs and separating outputs. In this case, the top and left sides of the circle represent the interactions between the processor and the inputs. On the right side, we have the resultant outputs. The solid lines represent flow. The dashed lines are a different matter.

The dashed line from the outputs to the controller consists of information about the current state of those outputs. That information, coupled with information the controller possesses about what those outputs are supposed to look like, serves as feedback. Any significant difference between actual outputs and intended outputs creates an error signal that leads to adjustments in the operations via the dashed line running from the controller to the operator. This is how control is exercised over execution of the process. Absent an error signal, the operator keeps on doing whatever it was doing – until it's time to stop. The operator stops performing those operations when the task is complete – when the desired result has been produced.

Similarly, the dashed line from the inputs to the controller also serves a control function. If the inputs aren't what they're supposed to be the operations might not be initiated or, if initiated, they will reflect an adjustment of some kind.

Control, then, is exercised over initiation, execution and termination of the process (what old-time industrial engineers used to call "Start-Run-Stop").

So there you have it. That's what lies inside the process box – or at least that's the way I see it.

The next section of this paper will delve into how to use the diagram above as a diagnostic and a design tool.

The Process Box Diagram: A Tool for Performance Diagnosis and Design

The situation that follows draws from a "process problem" I was asked to investigate when I was first employed at a major testing company.

A division director asked me to look into a situation in one of the programs operated in his division. The problem was stated simply enough: "The reject rate is too high." The operation in question was also simple enough: People wishing to take a certification test had to first register to take the test. The registration forms came in by mail, were batched and then scanned. Once scanned, the data from the forms were subjected to edits by the computer and, if the data failed those edits, the registration form was rejected from processing for resolution. The reject rate for a run of several batches of registration forms averaged between 60 and 70 percent. Of these rejected forms, roughly half had to be returned to the applicant. Clearly, that was a lot of costly rework, not to mention a lot of annoyed test takers.

A preliminary examination revealed there was nothing wrong with the process at the testing company. The high reject rate owed to the conditions of the registration forms submitted by applicants. In other words, the inputs to the testing company's registration processing operation were faulty. The registrants obtained a blank registration form from the company's testing bulletin. They filled it out and submitted it. The completed registration form was the output of the applicants' form-filling-out process and it was the input to the testing company's registration process.

The process box in which a blank registration form is transformed into a completed registration form is illustrated in Figure 3 on the following page.

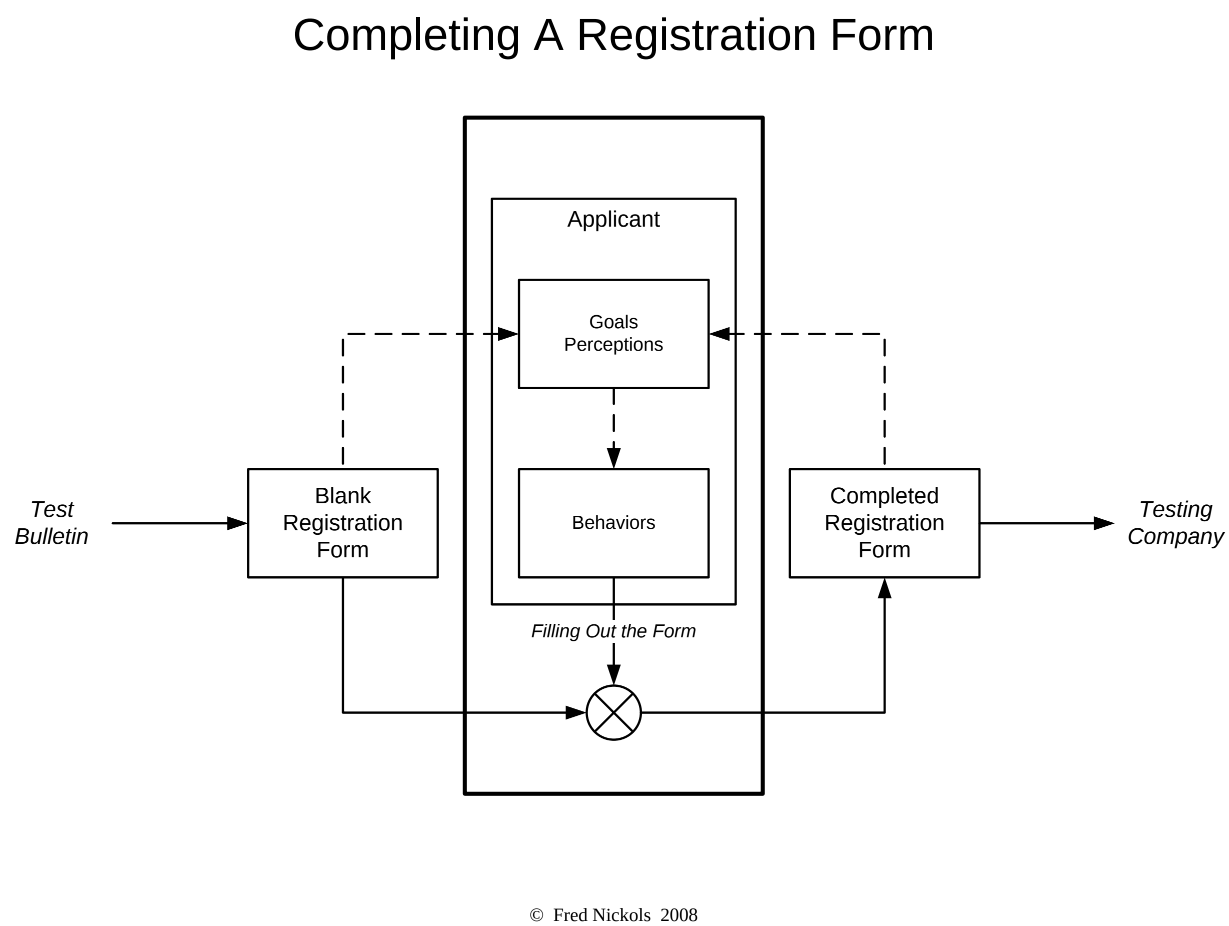

Figure 3

As Figure 3 shows, the input is a blank registration form, taken from the test bulletin provided by the testing company. The output is a completed registration form submitted to the testing company. And the process itself consists of various behaviors on the part of the applicant that can be summed up as "filling out the form." With the example above in mind, let's now look at the use of the diagram as a diagnostic and a design tool.

The basic diagram suggests numerous questions, all answerable from a diagnostic or a design perspective. These questions and their answers for the case in point are shown in callout boxes in the annotated version of the diagram in Figure 4 on the next page. They apply whether the processor in question is a human being, a machine, or a combination of the two.

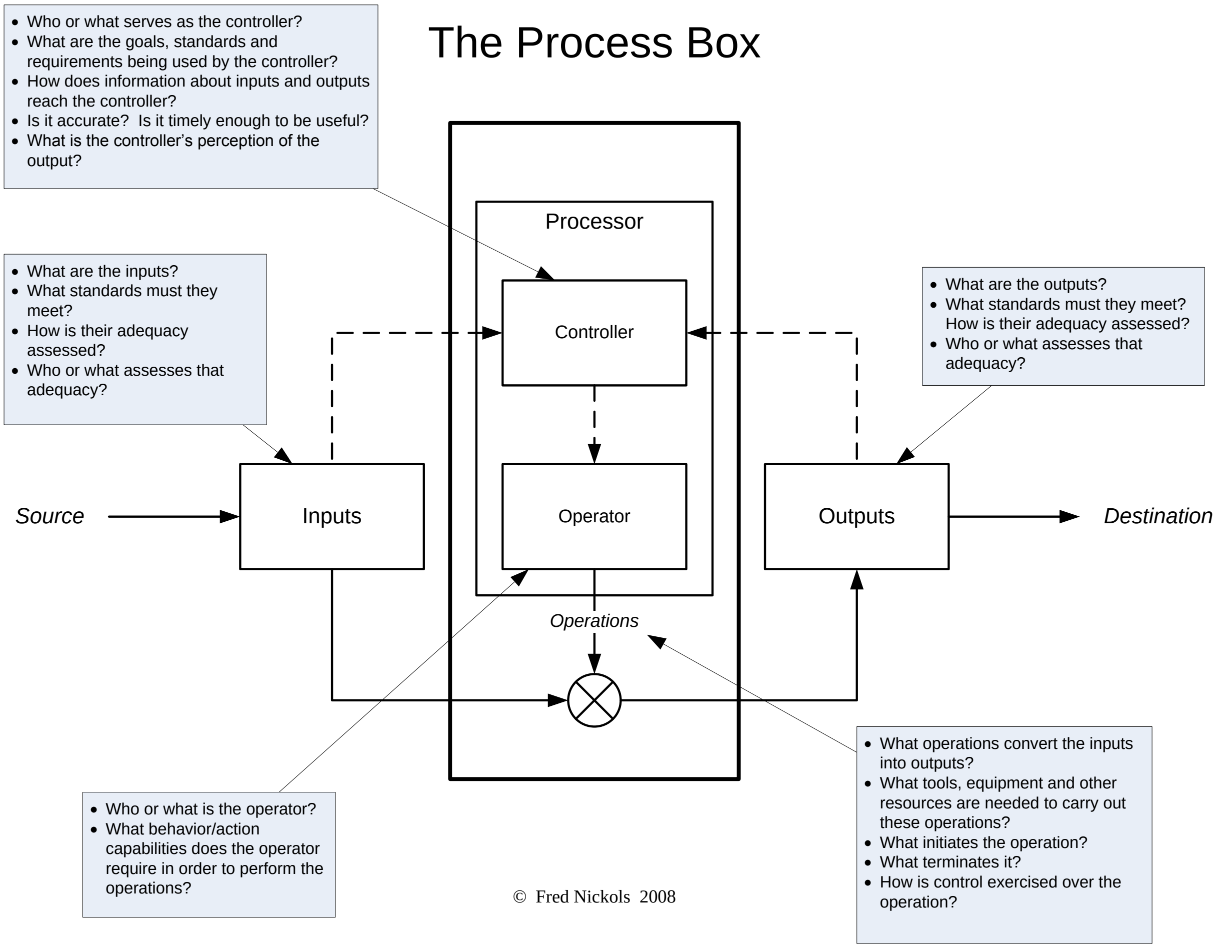

Figure 4

These same questions, with their accompanying answers for the case in point, are shown in the table on the next page.

Inside the Process Box – Diagnostic and Design Questions	
Question	**Answer**
Outputs	
What are the outputs?	A completed test registration form.
What standards must they meet?	“Clean and complete” (i.e., able to be processed “as is” by the testing company). All fields filled out and, where required, must contain a valid institutional code.
How is their adequacy assessed?	The testing company does so when the completed forms are received. The applicant does so at the time of filling out the form. But, other than some general notion about what is or isn’t acceptable when filling out a form or what can be inferred from the registration form itself, applicants have no solid basis for assessing its adequacy.
Who or what assesses output adequacy?	The applicant.
Inputs	
What are the inputs?	A blank registration form taken from the test bulletin.
What standards must they meet?	Blank or unused.
How is their adequacy assessed?	The applicant can see if the form is blank or not.
Who or what assesses input adequacy?	The applicant.
Operations	
What operations convert the inputs into outputs?	The applicant looks at the form to see what information is required. The applicant writes in the required information if it is known or available.
What tools, equipment and other resources are needed to perform those operations?	A No 2 pencil. An institutional code list in alphabetical order so applicants can look up an institution by name. Note: The list they have is in numeric order, used by the testing company to look up institutions by code number.
What initiates the operation?	The applicant’s decision to complete and submit the registration form.
What terminates it?	The applicant’s sense that the task is completed.
How is control exercised over its execution?	The applicant controls the filling out of the form.
Operator	
Who or what is the operator?	The applicant.
What behavior/action capabilities does the operator require?	The ability to read, write and recall or locate required information.
Controller	
Who or what serves as the controller?	The applicant.
What are the goals/requirements/standards used by the controller?	Ultimately, to become employed. That entails becoming certified. Certification hinges on passing the certification test and that, in turn, hinges on registering for and taking the test. In short, the applicant wants to register for the test. However, it is clear from the reject rate at the testing company that the applicants do not have a set of standards for completing the form that is consistent with the set used by the testing company. They cannot judge their own work.
How does information about inputs and outputs reach the controller?	The applicant is able to directly observe the registration form as it is being completed. However, many registrants also learn much later that the registration form they submitted has been rejected as unacceptable.
Is the information accurate?	There is no reason to believe the applicants’ perceptions of the forms they’ve completed are inaccurate. It is assumed they can see what they’ve produced.
Is the information timely enough to be useful?	Yes; there are no delays between writing and seeing what has been written.
What is the controller’s perception of the output?	This is probably best summed up as the applicant concluding, “That’s the best I can do. I hope it’s good enough.”

From a diagnostic perspective, the processor in question (the applicant) was not able to perform the operations necessary to convert a blank registration form into a properly completed one. The chief reasons were that (1) the applicants lacked the standards necessary to judge the adequacy of the forms they completed and (2) they did not have a suitable resource for completing the institutional codes portion of the form.

Now let's look at the use of the diagram in design.

In order for the conversion of an input into a specified output to occur, the processor must be able to perform the necessary operations and, equally important, judge the adequacy of the output produced. In this case, judging the adequacy of the completed registration form entailed ensuring that all fields were properly filled out and that the institutional codes listed were valid. A useful tool for identifying the code associated with a given institution would be an alphabetically organized list of institutions with accompanying codes. It is also the case that the form must be completed and submitted in time to result in successful registration for a preferred test administration date and location. These key requirements were unknown and/or unavailable to the applicants.

To make a longer story much shorter, applicants were subsequently provided with revised instructions that indicated and emphasized the importance to them of ensuring the registration form was "clean and complete," with connections clearly drawn to their goals of becoming certified and then employed. The instructions included an example of a completed registration form with all critical fields highlighted. The applicants were also provided with a revised institutional code list, one organized in alphabetical order. In terms of what is "inside the process box," the processor (applicant) was enabled to perform the operations required to transform the input (blank registration form) into an acceptable output (correctly completed registration form). Shortly thereafter, the reject rate plummeted from an average of 60-70 percent to less than 9 percent, dropping this particular issue to the bottom of the division director's priorities.

Recap or So What?

The "Inside the Process Box" diagram and the accompanying questions provide useful guidance to anyone seeking to analyze and understand problems of non-performance. They are useful from a diagnostic and a design perspective; they help understand the current state of affairs and why it exists, and they help determine what should be the case. You can create a table of questions like the one shown above – without the answers – and use it to gather and record answers when looking into problems of performance.

About the Author

Fred Nickols is a seasoned (that means "old") consultant, writer and once-upon-a-time executive whose interests lie mainly in the areas of systems, organizations, people and performance. He maintains a web site where his articles are available to anyone with an interest in them (www.skullworks.com). He still consults, albeit occasionally, and information about his services can be found at www.nickols.us. He may be reached by e-mail at fred@nickols.us.