Policy – Process – Procedure: What's in a Name?

Charles Wilson, Principal Engineer, Cybersecurity Development Lifecycle Practice

11/3/20 10:47:00 AM

Category: security-governance

Tags: security, cybersecurity, autonomous vehicles, governance, policy, process, procedure, SDLC, ISO 12207

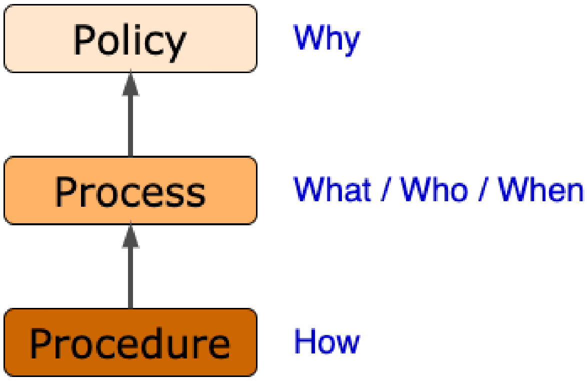
In upcoming posts, the terms: **policy**, **process**, and **procedure** will be used extensively. If you aren't involved in governance, these terms may strike you as identical or at least very similar.

To get the most out of our upcoming posts, understanding their distinct meanings and interplay is critical. In this post, we'll explore their meaning, use, and why we care.

Note: Links to more formal definitions are provided at the end of the post [1, 2, 3].

Paint Me a Picture

We can visualize the relationship and characteristics of the terms as follows:



As can be seen, they form a strict hierarchy. They also answer distinct questions. Now let's flesh out how they're defined and show how they work together.

Policy

The term **policy** refers to any formally documented and corporately approved principle, or set of principles, used to guide the decision-making process. It is the foundational rationale for why things are done the way they are.

That policies are both formally documented and corporately approved is a critical aspect of establishing the basis for certification. This is the case because policies are the root of all decisions made.

The formality of documentation allows us to point to this root. This is not to say that all decisions are made based upon policies, but that for the purposes of certification it is insufficient to base decisions on *ad hoc* or tribal policies, since these cannot be shown to be consistently applied.

An example of a cybersecurity policy [4] might begin:

> *As part of our operations, we need to obtain and process information. This information includes any offline or online data that makes a person identifiable such as names, addresses, usernames and passwords, digital footprints, photographs, social security numbers, financial data etc.*
>
> *Our company collects this information in a transparent way and only with the full cooperation and knowledge of interested parties. Once this information is available to us, the following rules apply.*

The main takeaway here is that a policy is a statement of intent and principle.

Process

The term **process** refers to any set of formally documented and corporately approved, structured activities that lead to a business goal (such as the creation of a product). Processes provide the framework by which policies are implemented. They tell us what is to be done, who is doing it, and when (relative to other processes) it will be done.

A good example of a set of processes is the SDLC (ISO 12207). Each process within the SDLC contains a set of activities supporting that process. These processes, taken collectively, support the policy of having a formal approach to software development. They do not, however, detail how the process is implemented. That is the domain of procedures.

Procedure

The term **procedure** refers to a sequence of specific, well-defined steps, which when executed, lead to the implementation of a process. It deals with how processes are accomplished.

Procedures produce artifacts. They do this by consuming identified inputs, utilize explicit tools, and generating specific outputs. Each step within a procedure can be scheduled, assigned, time-boxed, reviewed and audited.

Multiple procedures can implement the same process. An example of this is the creation of software library. The sequence of steps is dependent upon the language, development platform, target platform, build system, etc.

Your, Mine, and Ours

In an organization where certification is not necessary, these three tend to blur together. The nice way to describe it is organic. The problem is that it doesn't scale and it doesn't share well.

If you are developing a product subject to certification, you must show that you have policies and processes that support the case to be made for certification. There's little concern regarding procedure on the part of the certifying body. They're still needed though.

A second aspect is that when you need to scale development beyond a group or even beyond your company (collaborating with suppliers), it is very helpful to have shared (or at least compatible) policies and processes. Having cleanly separated policies, processes and procedures makes this much easier. Each organization is free to create custom procedures to implement common processes.

Room to Grow

It's important to recognize that the policies, processes, and procedures within an organization address the needs of multiple groups. This may be the C-suite setting the tone via policies, project management coordinating various aspects of product development via processes, or development teams consuming and producing artifacts by application of procedures. The three work together to provide order and structure to the product development lifecycle.

A final point motivating the explicit adoption of a **policy – process – procedure** pattern relates to maintenance. As the product or organization evolves, additional policies, processes, and procedures can be added or modified. Having this multi-tiered framework in place simplifies this.

More to Come

Hopefully, this post as provided a reasonable grounding in what policies, processes, and procedures are and how we use them. With them established, we can proceed to topics where they are applied. In my next post, we'll explore the product development lifecycle, where these come into play.

References

1. **Policy**
   https://en.wikipedia.org/wiki/Policy
2. **Process**
   https://en.wikipedia.org/wiki/Process
3. **Procedure**
   https://en.wikipedia.org/wiki/Procedure
4. **Company Data Protection Policy**
   https://resources.workable.com/data-protection-company-policy
5. **ISO/IEC 12207 Systems and Software Engineering – Software Life Cycle Processes**
   https://en.wikipedia.org/wiki/ISO/IEC_12207