

Procedure Methodology Explained

Please review the following material, which serves as a guideline for documenting our internal operating procedures and processes. This information serves as a quick introduction and is by no means exhaustive. Suggestions for improvement are welcome.

Our procedures must be meaningful by being useful. How we write our procedures affect their usability, which means that our audience must be able to:

1. *Find* what they need
2. *Understand* what they read
3. *Apply* what matters to accomplish the task(s) at hand.

Creating useful procedures—that's our objective!

So, when writing procedures,¹ let's choose one methodology and apply it consistently.

This document's material offers one way to develop meaningful and useful procedures.

Figure 1 depicts the proposed structure of our procedures, and the following text describes the sections' content.

Brief Description (Required)

Write a paragraph or two that describes the procedure.

Roles (Required)

Briefly address questions similar to the following:

- Who are the people, systems and applications/tools involved in this procedure?
- What critical dependencies exist?
- Where do the responsibilities lie?

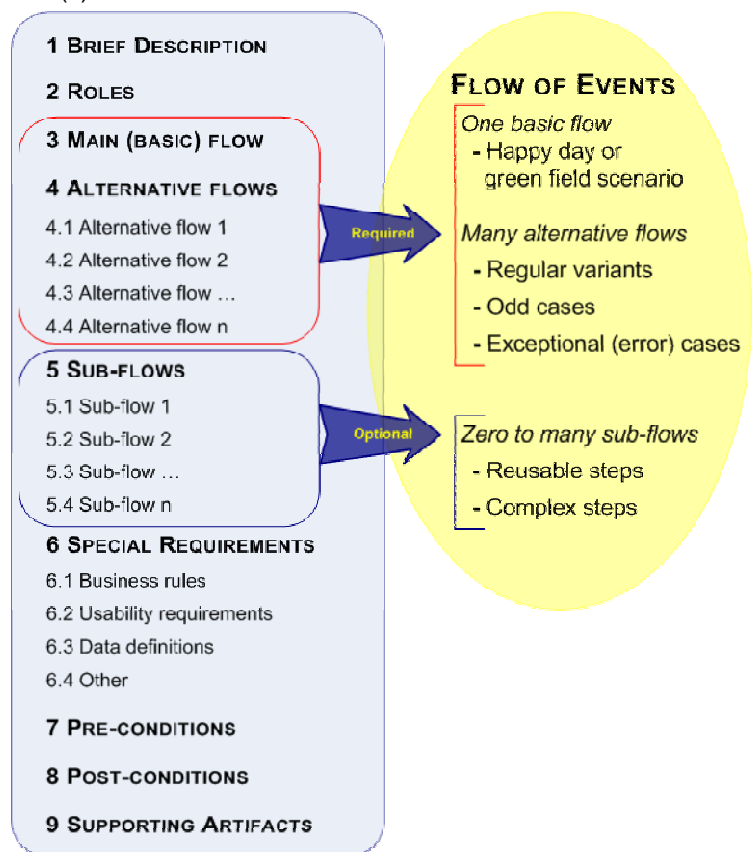


Figure 1 Procedure's document structure

¹ Procedures are step-by-step instructions that explain how to perform tasks. Four types of procedures exist: (1) *Single-step* procedures are standalone instructions that contain only one distinct user action or step; (2) *Multiple-step* procedures are standalone instructions that contain more than one distinct user action or step; (3) *Super procedures* contain other procedures; and (4) *Sub-procedures* (sub-flows) explain the steps in a super procedure.

Flow of Events

The flow of events consists of the main flow, alternative flows and sub-flows. The procedure's complexity dictates whether or not sub-flows exist in the procedure. Use the flow of events to describe how to reach the post condition from the pre-condition.

Main Flow (Required)

The *main flow* must exist. The main flow is the scenario where everything works as expected, from beginning to end. It's a "happy day" because the normal, straight-forward steps worked perfectly.

Recommended Practices

1. Structure the flow into steps.
2. Number and title each step: 1, 2..., n.
3. Describe each step using one to three sentences.
4. Do not refer to alternative flows in the main flow.
5. Use references, referring to titles instead of numbers.

Alternative Flows (Required)

Multiple *alternative flows* must exist, even with single-step procedures. Alternative flows extend the basic flow to allow for regular variants, odd cases and exception scenarios.

Recommended Practices

1. Identify the alternative flows.
2. Structure the flow into steps.
Divide complex flows into either additional alternative flows or create sub-flows.
3. Number and title each step: A1, A2..., An.
4. Use references, referring to titles instead of numbers.

Sub-Flows (Optional)

Sub-flows (subsections of other flows) are **optional** and depend upon the procedure's inherent complexity. Sub-flows explain and support the steps in the main and alternative flows, making these flows easier to read.

Recommended Practices

1. Specify only one thing, keeping the structure flat if possible.
2. Number and title each step: S1, S2..., Sn.
3. Use references, referring to titles instead of numbers.

Special Requirements (Optional)

Provide a textual description that reflects all nonfunctional² requirements on the procedure that are not considered in the flow of events.

² *Non-functional requirements* specify the qualities and constraints to which the system must conform: usability, reliability, performance and system support. *Functional requirements* relate to the system's required behavior: The system's input/output behavior is an example.

Pre-conditions (Optional)

Get the user(s) and the system(s) to the right starting point: Tell us under what conditions the procedure is applicable. The procedure can begin only when the precondition is true.

Precondition Example: The communication channel to `system x` is open and available for use.

Therefore, the procedure cannot begin unless the connection is available. The precondition must be true in order for the procedure to begin.

Post-conditions (Optional)

Use a post condition to define what the procedure is supposed to achieve. Convey the required system state when the procedure ends.






Note: Don't go looking for post-conditions. Describe meaningful—true **and** valuable—conditions. For example, stating that the administrator must be alive and that the system must be running is true but not valuable.

Supporting Artifacts

This section contains:

1. A simple, textual glossary that defines common terms specific to this procedure.
2. Procedural Diagram (Activity)

Table 1 Activity Diagram's Legend

Element	Meaning
	Beginning State—represents the event that starts the procedure's flow of events.
	Activity State—represents the activity's or step's execution within the flow of events. This is where you show the sub-flows and sections of the basic and alternative flows.
	State Transition—shows the activities' order. The activity's completion triggers the state transition.
	Decision Points—represents the procedure's extension points, allowing you to show the procedure's alternative flows.
[CONDITION]	Guard Conditions—control which flow(s) follows once the activity is complete.
	End State—shows where the procedure ends.

3. Supplementary Specifications (Optional): Legal and regulatory requirements are an example of summary specifications.

The glossary and procedural diagram are required.