

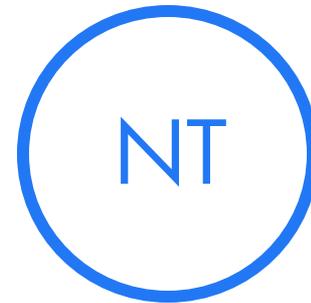


Service Desk
Troubleshooting Skill Manual

Deductive Lab



Analysts often get asked how the problem got fixed and not how the problem got identified.



TROUBLESHOOTING GOAL:

Test every suspect component until
problem cause is identified

What makes a problem more challenging to troubleshoot than others ?



2 components



100+ components

It all depends on the number of components.



20 components



100+ components

The higher the number of components or potential faults, the harder it is to troubleshoot.

First troubleshooting objective:

Locate the fault



Second troubleshooting objective:

Identify the fault



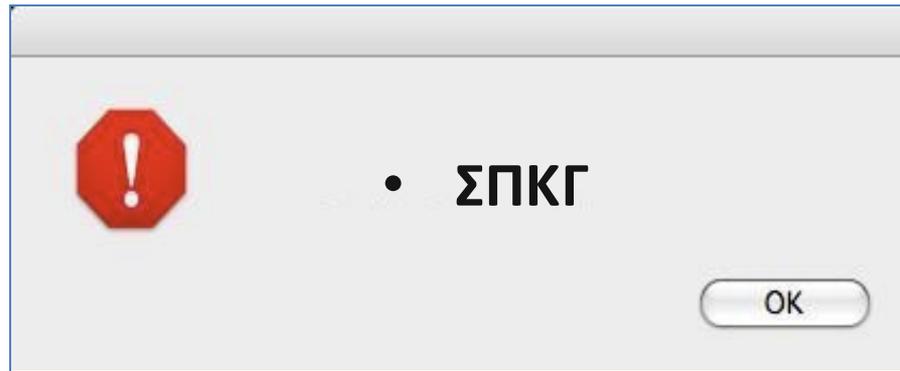
How do you troubleshoot?

For components with physical boundaries, troubleshoot visually.

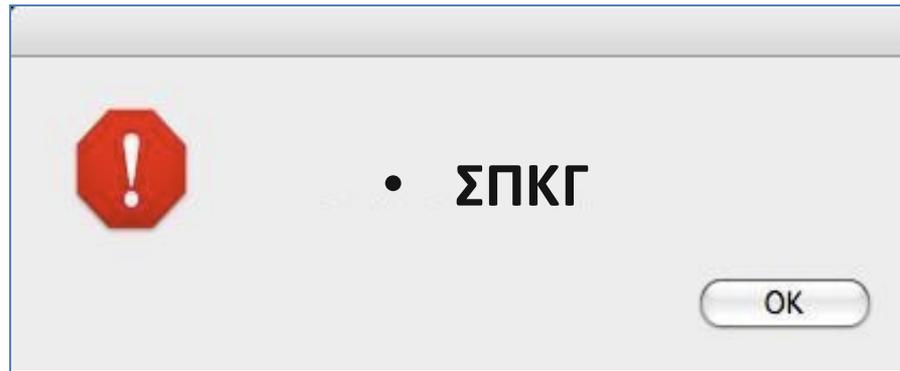


Let's define:
Troubleshooting

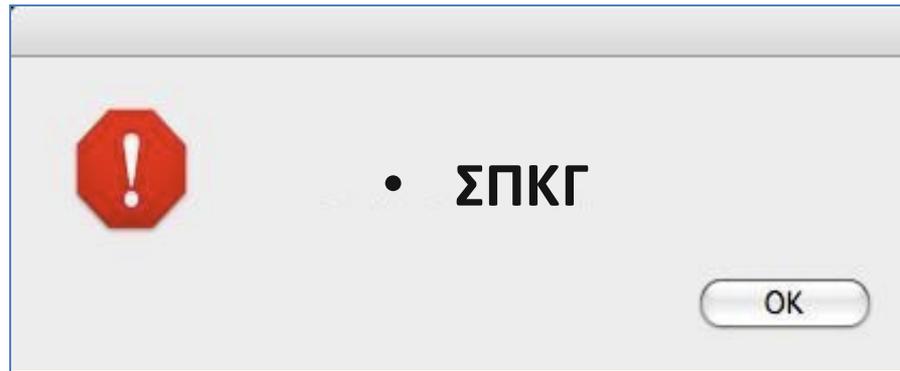
Problem X



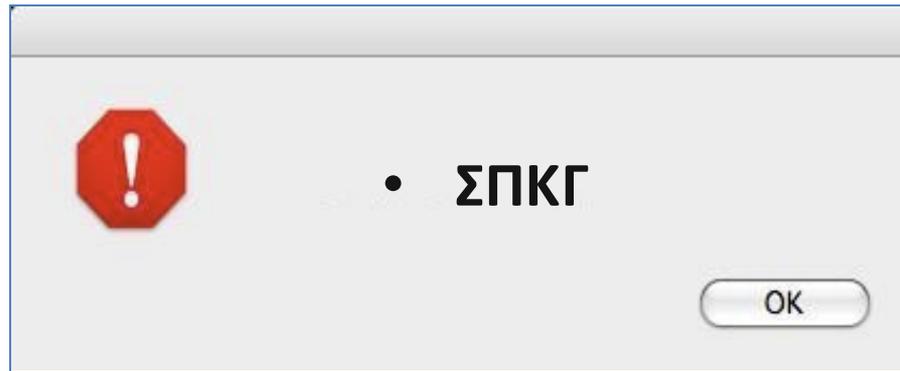
Unknown and undocumented technical issue



Replacing or reinstalling every component until problem X is resolved is **NOT** troubleshooting.



Applying known or popular fixes with the hope of fixing problem X is **NOT** troubleshooting.



Troubleshooting is the use of **technical deduction** and **logical isolation** to identify the cause of problem X.

Let's define:
Technical Deduction



Technical Knowledge + Deductive Reasoning =
Technical Deduction

Without **technical deduction** troubleshooting will
always start at:



“Is the power ON?”



Analyst with deductive skill, can **minimize** majority of suspect components just by asking questions.



Through deductive reasoning, an Analyst also can **locate** problem cause on any given problem.

Technical deduction is a required skill for any product support or help desk analyst.

Let's define:
Logical Isolation

Isolation Definition:

To set one or a group of technical components apart from others.



Why isolate?

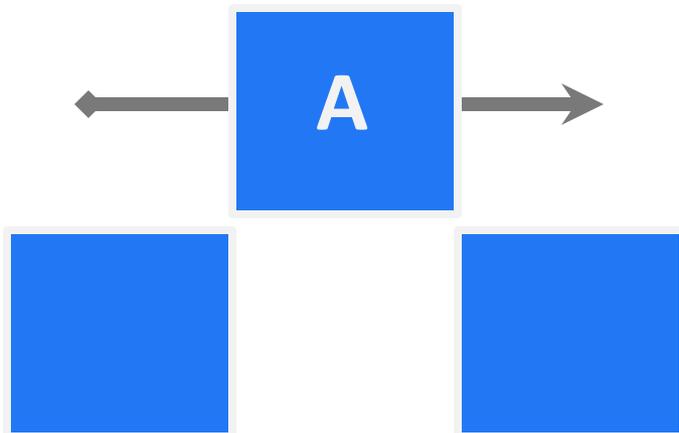
To test the component's functionality.

Individual components with physical boundaries can be easily isolated and tested.

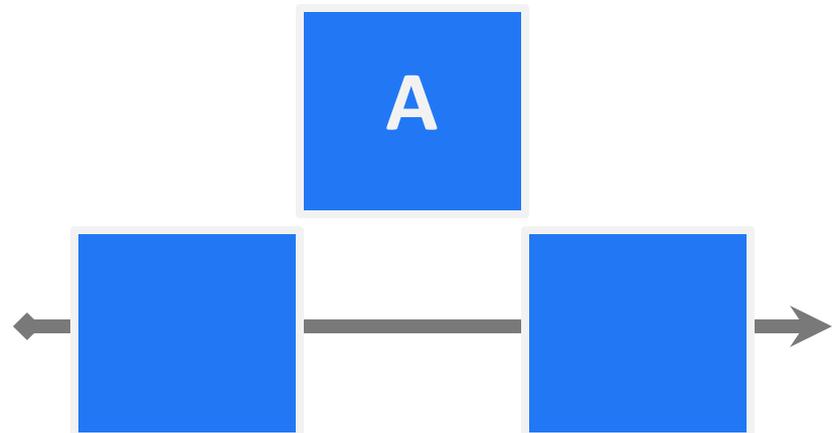


Types of Isolation

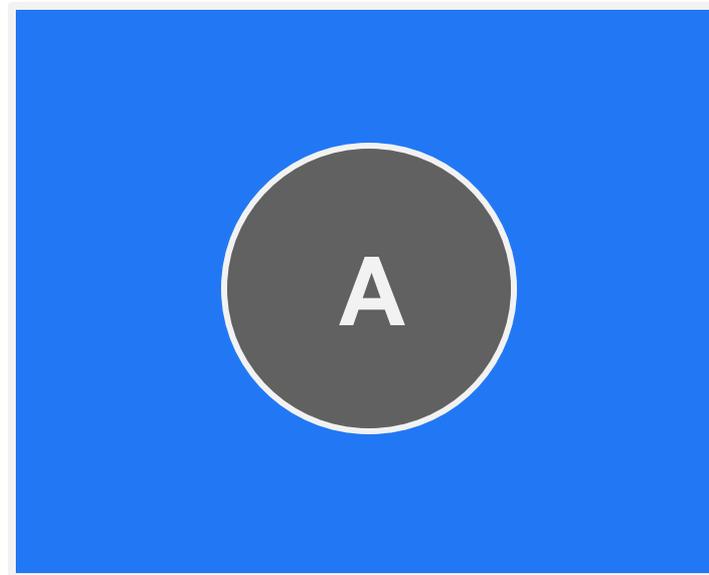
Testing A by
INCLUSION



Testing A by
EXCLUSION



Component with no physical boundaries needs to be isolated logically.

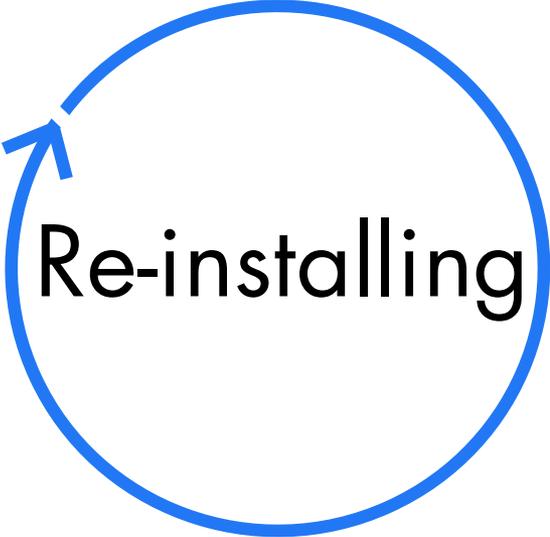


The faster one can isolate and test any given suspect component, the faster one can identify the problem cause.

#faults isolated / time =

Deductive Efficiency

Without **isolation skill**, troubleshooting will always start at:



Re-installing every component

...until the problem is fixed.

Let's define the methods
in: **Logical Isolation**

4 methods in logically
isolating faults:

simplify
shorten
compare
error for error

[simplify]

Process of excluding a complex component and substituting it with another simple or basic component.
“Is there a substitute or alternative core component?”

Example: wireless router to LAN cable; production server to test server; setting application setting to plain vanilla mode

[shorten]

Process of minimizing or shortening a process by excluding unnecessary components involved in a process. “Can this component be excluded out of the process without altering the overall process goal?”

Example: network printing process to local printing; process of remoting to an application on a cloud to accessing an application direct on a server console

[compare]

Process of comparing problem components with a known working component. “Is there a working model object to compare this with the problem object?”

Example: side by side configuration comparison; comparing application performance between a model hardware with the problem hardware

[error for error]

Process of injecting popular error into the target fault. The sequence of the expected error compared to the original error will help determine the location or identity of the fault. "Is there documented and reversible error that can be applied to this fault?"

Example: intentionally entering an incorrect password, which came first "wrong credentials/password " or original error ?

Logical Isolation Objective:
Identify Problem Cause

With isolation skill, one has the ability to **reduce** the number of suspect components to half just by asking deductive technical questions or a technical isolating task.

With genius isolation skill level, one can **identify** the problem cause or fault just by asking a series of deductive technical questions.



Deductive Lab

noel@deductivelab.com