itSM Solutions® DITY™ Newsletter Reprint

This is a reprint of an itSM Solutions® DITY™ Newsletter. Our members receive our weekly DITY Newsletter, and have access to practical and often entertaining articles in our archives. DITY is the newsletter for IT professionals who want a workable, practical guide to implementing ITIL best practices -- without the hype.

become a member

(It's Free. Visit http://www.itsmsolutions.com/newsletters/DITY.htm)

Publisher
itSM Solutions™ LLC
31 South Talbert Blvd #295
Lexington, NC 27292
Phone (336) 510-2885
Fax (336) 798-6296

Find us on the web at: http://www.itsmsolutions.com.
To report errors please send a note to the editor, Hank Marquis at hank.marquis@itsmsolutions.com
For information on obtaining copies of this guide contact: sales@itsmsolutions.com

itSM Solutions®
IT Experience. Practical Solutions.

Subscribe PDF Download Back Issues

Vol. 2.49 DECEMBER 20, 2006

How to Classify Incidents

DITY Weekly Reader

The workable, practical guide to Do IT Yourself

Most Service Desk staff (those performing Classification and Initial Support) will not know the cause of an Incident until the call is closed. So how can they identify the problem? The answer is that they can't and don't have to...

hank MARQUIS

Articles
E-mail
Bio

By Hank Marquis

How can one implement Incident classification? This is perhaps one of the most common questions that comes up from those trying to establish Incident Management based on the IT Infrastructure Library® ITIL ®.

According to ITIL the goal of Incident classification and Initial support is to:

- Specify the service with which the Incident is related
- Associate the incident with an SLA
- Select and/or defines the best specialist or group to handle the Incident
- Identify the priority based upon the business impact
- Define what questions should be asked or information checked
- Determine a primary reporting matrix for management information
- Identify a relationship to match against Known Errors or solutions.

Thus, Incident classification exists primarily to classify incidents in order to provide initial support. Initial support means proper analysis, evaluation and if required, routing. Classification is neither to determine root cause nor technical causes of the incident.

This single observation, that Incident classification is not to identify problems but rather guide workflow causes a tremendous amount of angst. The problem compounds when vendors promote classification schemes designed for knowledgeable technicians and not service desk agents.

The basics of classification are presented in previous articles (see below for links). In this article I want to explore the issues behind the actual classification hierarchy itself, which is where most practitioners experience problems.

Based on my experience helping to design classification systems the following compares and contrasts two different classification schemes, and provides a model that truly reflects ITIL practices.

Door Number 1—Category/Type/Item

Many IT Service Management tools that offer Incident management automation use a simple Category/Type/Item (CTI) for classification. CTI is a three-tiered approach of defining "Category", a "Type" associated with the "Category", and an "Item" associated with the "Type". One popular approach suggests that Category and Type be "nouns", and Item be a "verb".

This type of scheme yields classification taxonomy as follows (using CTI taxonomy):

category noun (Database) | type noun (Oracle) | item verb (Upgrade)

Thus, after determining the inquiry is an Incident (and not an RFC), and deducing that the Incident relates to an Oracle database requiring an upgrade the Service Desk staff would then code the Incident as Database | Oracle | Upgrade.

However, the CTI approach can limit your effectiveness because there are some not-so-subtle issues with its logic. CTI works well when the work required is known, as in this example. But CTI quickly becomes problematic when the workflow is not well known.

For example, how might a Service Desk agent know the "Database" category has a related type of "Oracle"? More importantly, what if there were multiple "Types" of Database; Oracle, SQL, mySQL, and Access for example? Which one would the Service Desk agent choose?

The extra investigation and diagnosis required to troubleshoot the Incident to complete the CTI classification is precisely the problem with the CTI approach -- it complicates data collection and combines Classification and with Investigation and Diagnosis, and confuses the purpose of Initial Support.

The reason is simple: CTI assumes a technical understanding of the causes of Incidents, and most Service Desk staff (those performing Classification and Initial Support) will not know the cause of an incident until it progresses through the Investigation and Diagnosis activity, and perhaps until closed.

In other words, for Service Requests CTI is fine, but for Faults CTI can become problematic when used by non-technical agents. Clearly we need another approach that is less technical, and more flexible.

Rethinking CTI

Lets go all the way back to what exactly is an Incident. ITIL defines an Incident as:

> "Any event which is not part of the standard operation of a service and which causes, or may cause, an interruption to, or a reduction in, the quality of that service."

This is a pretty large definition that covers two broad types of work:

- faults
- requests for new or additional services

Service requests encompass an additional level of detail. Examples of service requests include:

- questions about using services (e.g., application queries)
- routine actions (e.g., password resets)

Additionally, the Service Desk, where Incident management begins, also collects Requests for Change (RFCs.) While an RFC is not a type of Incident, the Service Desk has to be able to identify which handling action is needed.

This complicates classification a bit, since now we have to determine if the inquiry is an RFC and not an incident; and if an Incident which of three types of Incident it represents: Fault, Service Request (for routine action usually back-to-back with a Standard Change), or an Application Inquiry (how to use an application.)

Each of the possibilities will take a different path through the IT organization. This makes the first entry in the classification taxonomy a type (e.g., path through IT) and not a category.

The Service Desk has to be able to separate user inquiries into one of these four bins and then handle each appropriately. Now you begin to see why classification this is one of the most frequently asked practitioner questions, and why CTI may not be quite right for everyone approaching Incident classification.

Door Number 2—ITIL Classification

Classification and Initial support is for that reason – *initial support*. Initial support is determining what type of support the customer or user requires.

That is, the first entry in the classification taxonomy must indicate the type of work to be accomplished; it must clearly define how the IT organization must respond.

For these now obvious reasons, ITIL provides an example of this and labels the first element of its classification taxonomy as “Type”. There are just a few types based on the previous discussion regarding possible user inquires:

- Fault
- Service Request

Using a Type element establishes the basis for known work like RFC, Service Request, or fault; and allows differentiating lists. Examples of categories by type might include:

- service not available
- application issue preventing user from working
- Move/Add/Change to system
- disk-usage threshold exceeded
- system down
- automatic alert
- printer not printing
- configuration inaccessible

- request for information/advice/documentation
- forgotten password
- help user

Note how the main-category examples provided all report the issue in plain, non-technical, usage-based terminology. Users can only report symptoms of what they experience and request assistance in terms they understand.

After establishing the first element, “Type”, the next element “Category” changes based on the Type. For example, considering a Service Request for help and guidance about a software application a well formed classification might be (using ITIL taxonomy):

Service Request | Help User | Desktop Application

Now compare how CTI (noun-noun-verb) might write such a work request (using CTI taxonomy):

Software | Desktop Application | Help User

In comparison to CTI, note how ITIL taxonomy is noun-verb-noun. Also note ITIL clearly defines the work required of the organization (Service Request), helps the Service Desk agent or subsequent workers know what actions must occur (Help User), and finally what specialist should engage (Desktop Application).

Users only report symptoms relevant to their usage of the service, for example, unable to print from a Word processing application. This requires a noun-noun-adjective taxonomy of Type, Major Category, and sub-Category.

Consider another ITIL example, this time for a user with an application problem (using ITIL taxonomy):

Fault | Word Processing | Printer not printing

The practical result of CTI vs. ITIL classification is that of reduced classification tables with ITIL, and the ITIL classification schemes tend to be more “user friendly.” Finally, CTI almost pre-assumes an understanding of root cause and thus where to route the Incident, while ITIL aids routing without trying to diagnose root cause.

Those that favor the CTI approach are usually quite technical. They don't realize the value and limitations of a non-technical "front-end" to the process. These technical types often devise classification schemes which, in addition to including the expected resolution, wind up looking a lot like the support organization (using CTI taxonomy):

Packaged Software | MS Office | Macro Issue

or

Network Services | Cisco Router | Port locked

When a user calls it is not yet possible to know what the cause of the Incident is -- how would one know this is a "Network Services" or "Packaged Software" issue? In contrast, virtually everyone can talk to a user and determine if the Incident is a fault or a service request; determine which IT service, system or application is in question; and describe what the object of Investigation and Diagnosis ought to be.

In other words, it is less likely to mix Investigation and Diagnosis objectives with Classification and Initial Support objectives when approached from a ITIL perspective than an CTI perspective. This makes CTI difficult in many situations.

On the other hand, the ITIL approach has flexibility, and assumes that additional data (root cause, Configuration Item identification, etc.) come later during Investigation and Diagnosis, and the only goal of classification is to develop a clear understanding of the issue the user is reporting.

Thus, the ITIL method for classification is a "better" choice for most.

Summary

Classification schemes and their strategies for establishing types and categories will vary from organization to organization. However they share some common goals:

- they should always be agreed between IT and the business
- they should always be agreed between IT groups and the Service Desk
- they should direct further analysis, evaluation and routing, not attempt to diagnose root cause
- they should be as simple and easy to use as possible
- they should view things from a user perspective, not from an IT organization or technology viewpoint.

Even with properly configured service management software many still struggle with Incident classification. Common problems include:

- Mixing the objectives of "Incident Classification and Initial Support" with those of "Investigation and Diagnosis".
- Creating classification schemes with too many entries, making it difficult for Service Desk staff to navigate and provide initial support.
- Classification that is too technical, causing service desk agents to guess when trying to convert user reported symptoms into technical taxonomy.
- Having a classification scheme that looks like an IT operation organizational chart.

These problems all reduce the value and effectiveness of classification. However, forewarned is forearmed! Being aware of the issues other practitioners face can make your own journey easier. Be sure to see the related issues of scripting and Incident classification do’s and don’ts as well.

--

Where to go from here:

- Subscribe to our newsletter and get new skills delivered right to your Inbox, click here.

- Download this article in PDF format for use at your own convenience, click here.
- Browse back-issues of the DITY Newsletter, click here.

Related articles:

- 9 Steps to Better Incident Classification explains how to establish Incident classification systems.
- Scripted Success for more on creating and using diagnostic scripts.