DITY™ Weekly Newsletter
itSM Solutions®
IT Experience. Practical Solutions
The workable, practical guide to Do IT Yourself
Vol. 6.27 • July 23, 2010

How to Classify Incidents

By Hank Marquis
Hank is EVP of Knowledge Management at Universal Solutions Group, and Founder and Director of NABSM.ORG. Contact Hank by email at hank.marquis@usgct.com. View Hank's blog at www.hankmarquis.info.

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...

How do you implement Incident classification? This is perhaps one of the most common questions that comes up when 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 a Service Level Agreement (SLA)
- 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
- Select and/or define the best specialist or group to handle the Incident

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 have been 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, not a Request for Change (RFC) or Service Request, 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 required a type called "Oracle?" More importantly, what if there were multiple "Types" of databases – for example, Oracle, SQL, mySQL, and Access? 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 Initial Support with Investigation and Diagnosis, which 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.

This type of classification usually occurs when a group of technology specialists determine (on their own) how routing of tickets would work if they could design a system that they would use, to be used by people who know what they know.

Of course, the problem here is that technical staff do not perform Classification and Initial Support; relatively non-technical Service Desk agents do.

In other words, for Service Requests where the workflow is obvious, CTI is fine. However, for Faults or where workflow is not known or obvious CTI can become problematic when used by non-technical agents. Clearly, we need another approach that is less technical, and more flexible.

Rethinking CTI

Let's 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, often handled at the Service Desk)
- Routine actions (e.g., password resets or Requests, often routed to IT operations and resolved via Standard Changes)

Additionally, the Service Desk, where Incident Management begins, also collects Requests for Change (RFCs) through the Request Fulfillment process. While an RFC is not a type of Incident, the Service Desk has to be able to identify them and handle them as needed, usually to route to Change Management.

This complicates classification a bit, since now we have to determine if the inquiry is a Request and not an Incident; and if an Incident, which type of Incident it represents – Fault or an Application Inquiry (how to use an application or system feature or function.)

Each of the possibilities will take a different path through the IT organization. This truth makes the first entry in the classification taxonomy a Type (e.g., path through IT to a support group) and not a Category (e.g., what must happen when it gets to the right group.)

The Service Desk has to be able to separate user inquiries into one of these bins and then handle each appropriately. Now you begin to see why classification 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 just for that reason – initial support. Initial support is determining what type of support the customer or user requires. Classification determines the initial support the customer or user requires and this means 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 (not who in the organization must respond.)

For these reasons, ITIL provides an example of this and labels the first element of its classification taxonomy as "Type." The Type entry describes the broad functional involvement required to support the customer or user.

There are just a few types based on the previous discussion regarding possible user inquires. The exact number of Types is to be determined, but should clearly represent the major course through the organization. Some examples include:

- Service Request
- Fault
- Technical Incident
- Help/Assistance

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

Service Request

- Move/Add/Change to system
- Password reset

Fault

- Printer not printing
- System down

Technical Incident

- Disk-usage threshold exceeded
- Automatic alert

Help/Assistance

- Request for information
- Assistance using application

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. Note the separate category for non-user reported Incidents – Technical Incident.

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 the ITIL taxonomy clearly defines the work required of the organization (Service Request, not a Fault), helps the Service Desk agent or subsequent workers know what actions must occur (Help User, nothing to repair), and finally what specialist should engage (Desktop Application). This very clearly communicates how the organization must respond.

Note how the CTI taxonomy looks more like an IT organizational structure than a definition of required support. This is a key failure when using CTI. It is easy to fall into this CTI trap if you lose sight of the fact that Classification and Initial

Support is only to understand the support required. Overloading classification with too much technical direction reduces the effectiveness of classification to improve workflow and IT efficiency.

Users only report symptoms relevant to their usage of the service, for example, "unable to print from a Word processing application." This requires a slightly different and more descriptive 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* | *Cannot print*

Note how the classification describes what the user cannot do, not what the agent thinks the support group has to repair. "Cannot print" is very different from "clear print queue." Try to avoid classification that gives direction or predicts the failure, focus instead on fully describing in plain words what the customer or user cannot accomplish.

The practical result of CTI vs. ITIL classification is that with ITIL you can have reduced classification tables, and the 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 who favor the CTI approach are usually quite technical. They do not realize the value and limitations of a non-technical "front-end" to the process. These technical types often devise classification schemes that, 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 you 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 more likely to mix Investigation and Diagnosis objectives with Classification and Initial Support objectives when approached from a CTI perspective.

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 does not exist to establish root case or predict technical resolutions but rather to enable Initial Support, and Initial Support determines the workflow through the organization. Classification necessarily becomes more refined as the Incident progresses and more is learned via the Investigation and Diagnosis activities.

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 a technical taxonomy;
- Having a classification scheme that looks like an IT operation organizational chart because it attempts to determine and then route to the correct support group.

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.