# Project Deliverable Report

## Internal Deliverable ID3.2 – Incident Management Procedure

<table>
<thead>
<tr>
<th>Workpackage</th>
<th>WP3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task</td>
<td>ID3.2 Incident Management Procedure</td>
</tr>
<tr>
<td>Date of delivery</td>
<td><strong>Contractual:</strong> 31-01-2007  <strong>Actual:</strong> 14-05-2007</td>
</tr>
<tr>
<td>Code name</td>
<td>Version: 1.0  Draft □  Final ✅</td>
</tr>
<tr>
<td>Type of deliverable</td>
<td>Report</td>
</tr>
<tr>
<td>Security (distribution level)</td>
<td>Public</td>
</tr>
<tr>
<td>Contributors</td>
<td>Ruud Lemmers (LogicaCMG), Jean Koolen (LogicaCMG)</td>
</tr>
<tr>
<td>Authors (Partner)</td>
<td>Ruud Lemmers (LogicaCMG)</td>
</tr>
<tr>
<td>Contact Person</td>
<td>Ruud Lemmers (LogicaCMG)</td>
</tr>
<tr>
<td>WP/Task responsible</td>
<td>WP3</td>
</tr>
<tr>
<td>EC Project Officer</td>
<td>Hans-Jürgen Westhoff</td>
</tr>
<tr>
<td>Abstract (for dissemination)</td>
<td>Describes how incidents / bugs will be handled during the test phases of WP3.</td>
</tr>
<tr>
<td>Keywords List</td>
<td>Incident management, testing, incident, bug, Bugzilla</td>
</tr>
</tbody>
</table>

TENCompetence Project Coordination at: Open University of the Netherlands  
Valkenburgerweg 177, 6419 AT Heerlen, The Netherlands  
Tel: +31 45 5762624 – Fax: +31 45 5762800
Scope

The scope of this document is to describe how incident management is done during the testing and pilot phases.

The basis for this procedure is the ITIL (IT Infrastructure Library) standard commonly used in IT projects. For more information about ITIL visit [www.itil.co.uk](http://www.itil.co.uk).

Purpose

The goal of this procedure is to explain project members how to deal with incidents.

The goal of incident management is to repair the level of service as soon as possible and to guarantee the service levels of quality and availability as best as possible.

Definitions

An **incident** can be defined as “every event that does not belong to the standard service operation and that can lead to a disturbance or decrease of the quality of the service”. A **bug** is an error in a computer program. Incidents and bugs are similar, the difference is in their scope. Incidents can be triggered not only by computer programs, but also by things like mistakes in manuals, incorrect test documentation, insufficient service levels.

A **change request** is a formal request to change previously agreed demands of the system (i.e. functionality, behaviour, performance, …).

**Service requests** are requests of a user for 1st level support for information, advice, support or documentation. For the testing phases, 1st level support is provided by the developers. For the pilots, 1st level support is provided by the organisers of the pilots (UPF for Digital Cinema; SU for ICT Teacher Training).

It is more efficient to deal with change and service requests separately. The way they are dealt with is not described in this procedure.
Processflow incident management

The flow shows incidents are reported by project members, not directly by students. Reporting an incident is done by Registration in the defect tracking system (see the section Administration tool for detailed information about submitting incidents), which is done by either testers or pilot support members from UPF / SU.

An Incident Manager (a role for experienced LogicaCMG testers) checks as Intake if the required information has been entered and if the report seems justified (i.e. not a duplicate of another incident or caused by an incorrect test or …).

When the Intake and Registration are complete, the next step is Classification of the request.
Classification of the incident is based on three aspects:

1. **Impact**
   The consequence of an incident on the level of service.

2. **Priority**
   How important is it to resolve the incident quickly? This determines the lead time in which the incident should be solved.

3. **Effort**
   The amount of time required to solve the incident. The estimated effort is entered as part of the “Additional Comments”.

Based on these three factors the priority of the incident related to other incidents is defined. *Classification* is done by an incident manager.

If the request is a service request, the request is dealt with using the *Service request procedure*. If the request involves changing instead of fixing something, the *Change management procedure* is used.

It should be clear that detailed investigation and diagnosis of the incident is done by 2nd level support and not by the incident manager / 1st level support. 2nd level support is done by the code owner of the part causing the incident, which is either WP3 or an Aspect work package. In case an incident is caused by the configuration of the test environment’s server(s), it is assigned to the server’s administrators for resolving.

After assigning an incident to 2nd line support, the 2nd line support developer will *Analyze* and *Resolve* the incident.

All requests (incidents, service requests and changes) are finished by sending *Feedback* to the person who registered the request, to explain what has been done to handle the request.

**Administration tool**

Incidents are stored in the “Defect Tracking System” Bugzilla. Defect Tracking Systems allow individuals or groups of developers to keep track of outstanding bugs / incidents in their product effectively. For more information about Bugzilla, visit [www.bugzilla.org](http://www.bugzilla.org).

The primary goal for Bugzilla is registering bugs, this is why text below that refers directly to Bugzilla uses the term ‘bug’. Within TENCompetence testing, all incidents, not only bugs, will be registered in Bugzilla.

The Bugzilla installation for TENCompetence is at [http://wush.net/bugzilla/rulem](http://wush.net/bugzilla/rulem). Users can use their web browser to submit or update incidents on this site.
The template for entering incidents is Bugzilla’s entry form for a new bug:

To ensure a proper intake and resolution of an incident it is important to provide accurate and detailed information about the incident during registration. The sections below describe the main items to register. Details about all the fields from Bugzilla are provided on-line: [http://wush.net/bugzilla/rulem/page.cgi?id=bug-writing.html](http://wush.net/bugzilla/rulem/page.cgi?id=bug-writing.html).

**Summary**
One line description of the incident.

**Priority**
Define the level of impact, priority, the effort required to solve the problem and the relations with current problems or initiated changes.

**Description**
Personal information (only if the incident was found by someone other than the person entering the incident):
- The name of the person who discovered the incident.
- Contact information of the person who discovered the incident.
Description of the incident (as detailed as possible):
- The activities preceding the incident.
- The actual and the expected result.

It is important to describe this as detailed as possible, to increase the chance that the incident can be reproduced and to increase the chance of solving the incident.

**Initial State / Status**
The current status of the incident should be recorded as well. This provides an overview for the incident manager as well as for the project members.

The possible statuses and their transitions are as follows:
Resolution
The field remains empty as long as the incident has not been resolved. When filling in
a value, it has to be selected from a list. The available values and their meaning
(copied from http://wush.net/bugzilla/rulem/page.cgi?id=fields.html#status) are as
follows:

- FIXED
  A fix for this bug is checked into the tree and tested.

- INVALID
  The problem described is not a bug.

- WONTFIX
  The problem described is a bug which will never be fixed.

- DUPLICATE
  The problem is a duplicate of an existing bug. Marking a bug duplicate
  requires the bug# of the duplicating bug and will at least put that bug number
  in the description field.

- WORKSFORME
  All attempts at reproducing this bug were futile, and reading the code produces
  no clues as to why the described behavior would occur. If more information
  appears later, the bug can be reopened.

- MOVED
  The problem was specific to a related product whose bugs are tracked in
  another bug database. The bug has been moved to that database.

Additional comments
This field is only visible when editing an existing incident. It should be used for
things like:

- Extra information about the incident.

- Description of how the incident was solved.