

Defect Reporting And Resolution

L. S. Diehr
January 25, 2000

Overview

Defect reporting and resolution seems a simple thing to most people, but in reality, it can be a very complex and involved process when all of the perspectives are put into place. The process of resolving a defect and reporting the results in a traceable manner is one of the major points of the ISO quality standards, and is a major key to understanding how to achieve the hurdle of ISO 9000 certification.

A defect report may start out as a simple recognition of an anomalous behavior in a system - something out of the ordinary or unexpected. At this point, a report of the event should be filed. This report should then be analyzed to determine the severity of the suspected defect, then assigned for resolution, a fix developed, verified and finally released. This is a straight forward framework with easily understood steps that needs a little expansion so it can be implemented.

There are several goals involved in the process that must be satisfied.

Goal 1: Never have to fix the same problem twice

Goal 2: Keep everyone on the same page

Goal 3: Measure the effectiveness of the resolution process to identify opportunities for improvement

Filing A Defect Report

Anyone can report a defect – especially if they found it. Reports of defects, however, must contain a minimum of data in order to be useful:

1. What is the product being run. This may not always be exactly known due to the nature of the implementation – whether a menu is from the program the user invoked, or from some program invoked to deal with a screen.
2. What are the environmental conditions – did the system crash, was data destroyed, or a bad part produced -- what else is running?
3. Date – either the date the defect was found, or the date of the report (hopefully the same date as the occurrence). This will be used to measure the effectiveness of the defect removal process.
4. What actions were taken just prior to encountering this defect (there may be a cause/effect relationship)
5. Who is reporting the defect (such as an e-mail address) .

Analysis And Reporting The Analysis

Often it is best to use a multi-layered approach to this problem. The first layer is a simple filtering of repeat or simple problems. In the case of repeat problems, refer the repeat to the original. This provides traceability from the customer to the product functions. Out of the first layer of analysis, a defect is: rejected; assigned for resolution; or deferred and escalated.

- Rejection is according to strict guidelines (e. g. Repeat).
- Assigned for resolution if only a small change is required (within current project timing).
- Deferred defect reports must be analyzed at a deeper level to determine the cause and the amount of effort to affect the repair.

Making A Decision

The escalation process depends on someone being able to identify and characterize the defect with enough detail that management can assign the appropriate resources.

Escalation is the process of bucking the decision to higher and higher levels of management as the options and impacts of the proposed fixes impact customers. This process can go to the very highest level but for the sake of argument, there are three levels of escalation from the point it first arrives.

Level 1 – (easy to fix, can be contained within the current schedule)

Level 2 – A fix may (but not expected) impact the current development schedule.

Level 3 – The effort to fix is great, current schedule cannot contain the effort or implementing this fix will cause other functionality to be dropped from the product.

Defect Classification

Defects or Anomalies need to be classified from several perspectives:

1. From the user’s perspective (Severity):
 - Disastrous -- when encountered crashes or locks up system or loses data.
 - Important -- performs incorrectly or inappropriately.
 - Suggestion for improvement - minor/irritating problem for which there may be a work-around, or to report spelling, consistency or performance issues enhancement requests.
2. From the developer/analyst perspective (Impact):
 - Severe -- defect causes loss of data or causes system to become unusable.
 - Serious -- a defect that makes the functionality unreliable, inaccurate, or otherwise inappropriate and for which there *may* or *may not* be an appropriate work around.
 - Minor -- a solution is at hand or the fix is simple and apparent.
 - Repeat -- a defect that has been reported before, regardless of it’s classification.
 - Request -- a request to change the way the product works.
3. From management perspective (Priority):
 - Mandatory -- a statement that all resources available will work to resolve this problem with no impact to the scheduled release date of the next version. Functionality will be cut in preference to not fixing this problem.
 - Imperative -- a statement that while important, resolution of this defect will not slow the production schedule.
 - Optional -- this defect will be repaired if there are resources available to do so without affecting the schedule.

The specific words used in this classification system are not as important as that different words are used for each category of perspective.

Generic Process

Assign Repair — a manager or supervisor must assign the work of the resolution to an individual. The documentation from the analysis must accompany the defect report. If possible, additional materials can complement or verify the analysis. Items like design and requirements documents, past and related changes to the system or it's components, etc.

Develop Repair — Fix the software. Check modules out of Source Code Control System and make appropriate changes and unit tests. Check new source code back into the SCCS. Defined tests must include tests to detect the original problem and tests that show the defect removed. These tests will ultimately become a part of the normal regression testing of the system.

Verify Repair — After the development tests have been run, the repair must be integrated into the environment to be released to the customer. This new release must then be verified to assure nothing new has broken and all of the old stuff still works as expected.

Release Repair — The new validated system with the repair(s) is released and installed in the user’s environment.

The number of severity levels must be kept small so it is easy to tell them apart. Definitions need to be clear and some simple examples may be appropriate. Severity must be assessed from the customer perspective rather than the difficulty of any fix.

Keep Everything

It is important to keep the log of rejected and released reports for future reference. Patterns that emerge from analysis of these as a better understanding gained with more experience may lead to forwarding the report through the system, or may become the basis of a request for enhancement and made into a development project for incorporation into the product at a future time. This is a typical assignment for a Quality Assurance group.

Tracking The Points Along The Way

New

There must be a starting and ending point identifiable in the process. The beginning is at the point the anomaly is found. There are several audiences that could be making these reports: end users (customers), participants in the product test process, and those participating in the development process. Everyone has a perspective on a particular report, and a different level of background knowledge upon which to base expectations of the product. The customer will usually have the most naïve view of the internal interactions of the system and must be allowed to express that understanding rather than being forced to have a detailed technical view of the product. Those from the testing environment need to express their perspective in terms of specific tests performed and the expected results. Those involved in the development of the product have in-depth knowledge of the internal processing and interfaces of the system and must be able to express that perspective as well. This means that the each perspective needs to be a consideration when implementing any automated reporting system.

Released and Rejected

The ending point for a report must be clear, and there needs to be a way to get things out of the middle of the process that either don't belong, are duplicates, or are really enhancement requests. Two ending points are shown here - Rejected and Released. Rejected reports are primarily for those that cannot be attributed to the product, are suggestions for enhancement, or otherwise inappropriate to be classified as defects against the product. Repeat performers are really a special case as they go to show the critical nature of the defect. Those defects identified often should be considered more critical to the audience of users, hence move up in priority for

resolution.

Requests for enhancements are not reports of defects and the defect resolution system should not be a way to define or track the development of enhancements. This is a difficult policy to enforce on the outside world, and user requests should be encouraged. The reject pile should serve as a source for enhancement requests for planned definition, development and implementation.

Analyzed

Making a determination can be a long and complicated process. Usually, it is a simple process based on knowledge of the system and the kinds of problems reported in the past.

Analysis is aimed at identifying the root cause of the defect, and the information maintained should reflect that goal. For the purpose of tracking the development of the repair, the methods and formal documentation are not a concern beyond being able to identify the information for transmission to the next step and for reporting on the effectiveness of the system and robustness of the product.

Assigned

This is a state that deals with the commitment of resources to the resolution of a problem. Once the scope of the problem has been identified, a person or team can be assigned to make the problem go away. Depending on timing, resource requirements, technological impacts and other factors, the decision may have to be escalated. This is the point where defect tracking encounters configuration management, and the two should not be confused. Once a decision has been reached, it is appropriate to assign the resolution development to someone. Notification of the assignee is appropriate now.

Developed

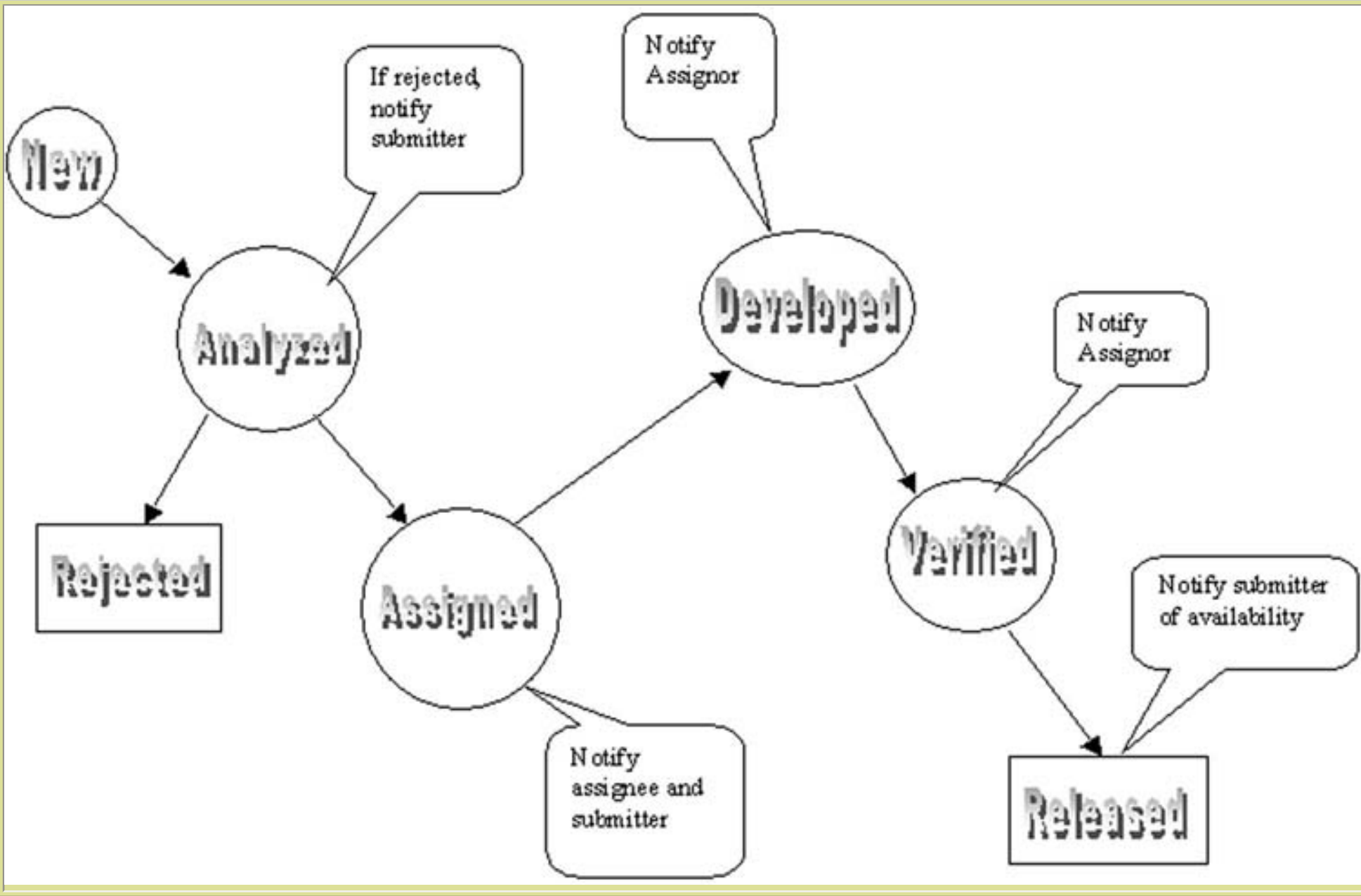
Between these Assigned and Developed are all of the processes involved in designing, implementing and low level testing of the fix. This is the point that the fix can be integrated with the system and an independent group used to verify the fix has indeed been successful. Notification of the testers and management may be appropriate at this point.

Verified

This is after the fix has been tested and it is ready for release in the product. Notification of the developer and managers of the status of the testing effort (passed or failed), and movement either back to the developers, on to be released, or potentially back for analysis if things are really messed up.

Released

There is nothing to do now but wait until the next release of the product can be generated and distributed. Notification of the submitter (and all those that submitted repeats) can now be made indicating when the fixed functionality will be available for use.



■ Figure 1. State Transitions and notifications for defect resolution:

Reporting Results

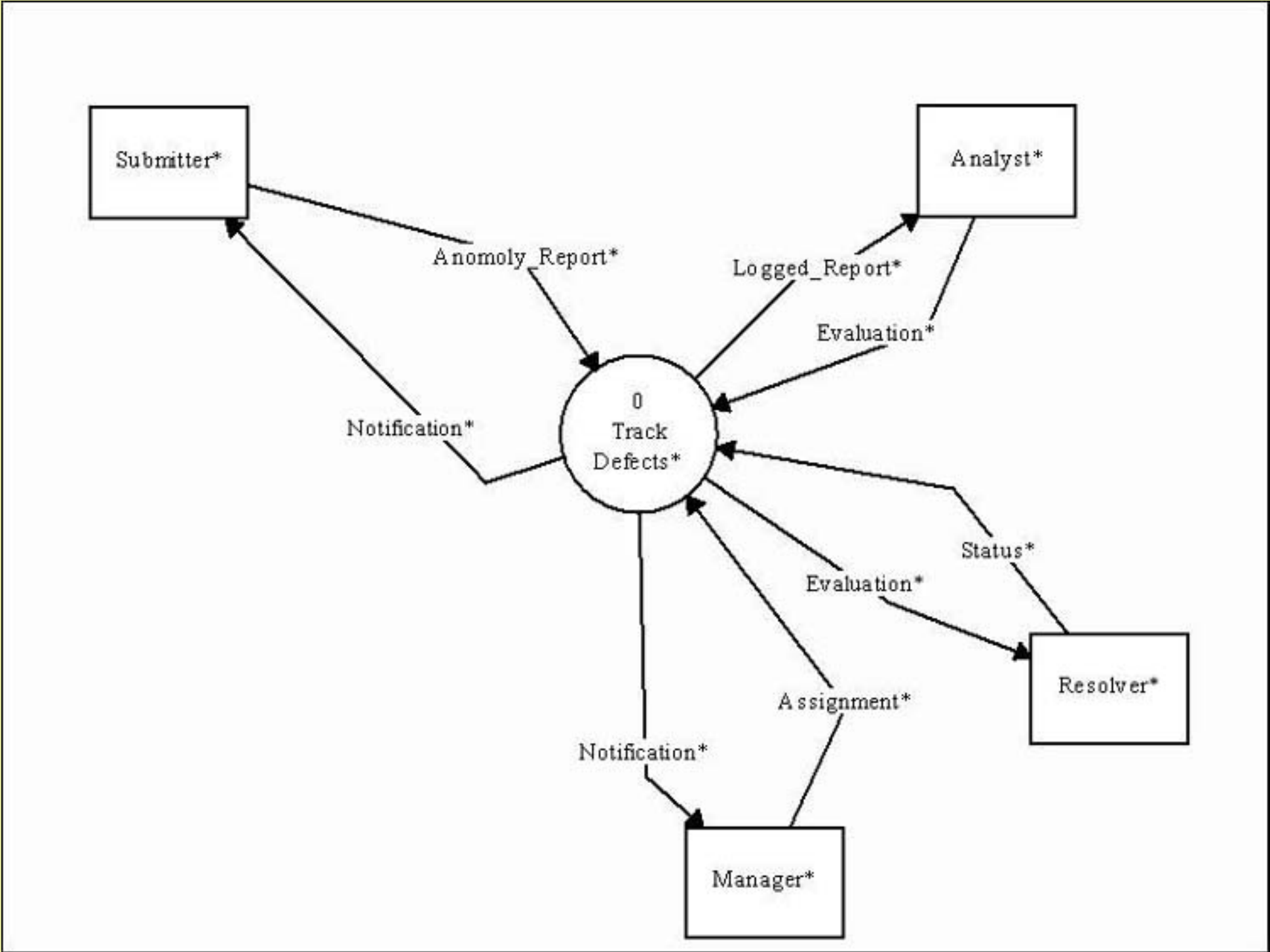
Effectiveness

Here is where the dates of the transitions become important. Measuring the effectiveness based on performance is critical to identifying bottle-necks, definition problems and other process related issues. If it is taking a very long time to analyze a problem after it is reported, or to make the assignment after the analysis, there may be a problem in the management philosophy surrounding the process. If the fixes are taking a long time, or there are a very large number of problems being reported, there may be some difficulty in either the development process that produced these problems, the priorities of the developers, or other factors that can only be expressed and exposed in this manner.

Notification

As the defect moves through the process, there should be notifications sent to interested parties. These notifications should be automatic and should indicate the transition being made. Determining who should be notified when the transitions occur can become a nettlesome affair, and often people don't want to have their mail pestered by automated notifications. An alternate approach is to use reports and to distribute them.

Information Flow



■ Figure 2. Information flow for tracking defects

[Contact Mr. Diehr by e-mail](#)