

# Defect Management

## 3 Components of Effective Defect Management Systems

Krishen Kota, PMP

Software development teams and software testing teams have numerous choices of defect management tools to help support their software defect efforts. Even though I am a big proponent of a particular defect tracking tool (AdminiTrack.com), selecting and utilizing an effective tool is really only part of an overall defect management system.

From a high-level view, defect management systems are made up of a combination of some defect management tools or tool and a defect management process. These two primary components work together to support each other. Ignore either one, and sub-optimal results can be expected. Below I will provide an overview of a typical defect management process. I will also list the key features to look for in an effective defect management tool. All of this information will be useful for you to review. But I would be remiss if I left out one of the most overlooked aspects of software defect management that goes beyond tools and processes, which I will also disclose later in this article.

## Defect Management Process

### High Level Steps in a typical defect management process:

The typical defect management process includes the following high level process steps. When implemented inside of a specific organization, each of these high level steps would have more detailed standard operating procedures along with policies to carry out the details of the process.

1. **Identification** - This step involves the discovery of a defect. Hopefully, the person discovering the defect is someone on the testing team. In the real world, it can be anyone including the other individuals on the project team, or on rare occasions even the end-customer.
2. **Categorization** - When a defect is reported, it is typically assigned to a designated team member to confirm that the defect is actually a defect as opposed to an enhancement, or

other appropriate category as defined by the organization. Once categorized, the defect moves on in the process to the next step which is prioritization.

3. **Prioritization** – Prioritization is typically based on a combination of the severity of impact on the user, relative effort to fix, along with a comparison against other open defects. Depending on the size and structure of the organization, the prioritization is often handled by a formal change control board. The priority should be determined with representation from management, the customer, and the project team.
4. **Assignment** - Once a defect has been prioritized, it is then assigned to a developer or other technician to fix.
5. **Resolution** - The developer fixes (resolves) the defect and follows the organization's process to move the fix to the environment where the defect was originally identified.
6. **Verification** - Depending on the environment where the defect was found and the fix was applied, the software testing team or customer typically verifies that the fix actually resolved the defect.
7. **Closure** - Once a defect has been resolved and verified, the defect is marked as closed.
8. **Management Reporting** - Management reports are provided to appropriate individuals at regular intervals as defined reporting requirements. In addition, on-demand reports are provided on an as-needed basis.

## Defect Management Tools

### Features of a defect management tool:

Following are the core features of a defect management tool:

- Provides a centralized repository for tracking defects across projects.
- Provides automated notifications of resource assignments.
- Ability to define defect resolution status in order to map back to your defect management process.
- Ability to provide management reporting, like the number of open defects grouped by various criteria such as open defects by project, severity, and priority.

Following are a few optional features worth mentioning when selecting a defect management tool:

- Ability to capture other items in addition to defects, such as customer suggestions and project-related issues. Items such as customer complaints or enhancement suggestions are often lost if not logged in a centralized system. If other tools are not already available, a defect management tool can be used to track these types of items as long as they can easily be filtered out or logically separated from defects.
- Ability to support internal and external teams. This feature provides the opportunity to involve external teams and in some situations, customers if appropriate.

Another consideration when selecting a defect management tool includes ease-of-use. Comparing a list of features is useful, but seeing how intuitive a system is to the people who have to use it provides a more concrete sense of how much training might be needed along with how well the system will be received.

## **The Missing Component of Software Defect Management**

Most discussions on the topic of software defect management focus on defect management processes or defect management tools. That is where most stop. There is an additional and often overlooked aspect which is more important than the specific defect management tools or defect management process being used. I consider it the important critical success factor in software defect management. I call it "Organizational Culture," which is comprised of the shared values, beliefs, and accepted norms of the people in an organization. But wait. Before you succumb to your sudden urge to tune out because this article just seemed to take a sharp turn towards some soft and fuzzy, executive management-level, buzzword-compliant rhetoric, please read on a bit more because there is a practical side of this that you may want to know. Getting even a crude understanding of how organizational culture affects software defect management could make the difference between success or failure on a high-profile project you are involved with.

From my experience, in many organizations, defects are considered something negative, for which blame should be assigned as a way of preventing similar defects in the future. This sounds perfectly reasonable, right? The problem is that just because something seems to make sense on paper doesn't mean it directly translates to the real world.

Here are a couple of questions to consider. If team members feel like any mistakes they make may be used against them in a performance review which could affect their compensation or job stability, are they less likely to make mistakes which would decrease the number of defects? Possibly. But I would ascertain that the side effects are much more negative than having lots of defects. You want to know what defects you have and you want to know about them as soon as possible.

As is commonly understood, the later in the process defects are identified, the most costly they are to fix. If defects are treated as negative things to be avoided, people will become less likely to disclose them, and may spend too much time on certain tasks in an attempt to avoid mistakes and subsequently defects. A hyper-focus on avoiding defects may be appropriate for certain situations, but for most business settings, an unrealistic focus on perfection can be a fatal flaw that leads to suboptimal performance, which can lead to outcomes which are far more negative than the defects some strive to avoid.

“Field experience and data from leading research in organizational culture shows that a large majority of organizational cultures are defensive in nature, where people are punished rather than coached on correct procedures when they make mistakes,” said Buz McOmber, President of Constructive Cultures, an Atlanta-based organizational performance consultancy. “Fearing negative consequences, people learn quickly to cover their own mistakes, blame them on others, or push the correction off as ‘not my job’. These behaviors push defect identification to later in the development process or, worse, to the production environment, significantly raising costs to the business and harming both operational effectiveness and customer satisfaction. Fear of negative consequences is intensified in tough economic times, increasing the need to identify these defensive behaviors before they make a difficult situation worse. For the Project Manager, it can be very difficult to regain the trust that’s lost when their team fears retribution for mistakes,” McOmber said.

If you are serious about having an effective defect management system, at a minimum you need to have a defined process, effective tools, and a culture that understands that defects are a simple by-product of getting work done in the real world. They are not mistakes that need to be avoided or covered up at all costs.

## **Conclusion**

A defect management system is made up of a combination of defect management tools or tool and a defect management process.

In addition, the effectiveness of a defect management system is influenced by the organizational culture it operates within. For most environments, it makes sense to utilize tools and processes that focus on the speed of identifying, tracking, and resolving defects. This provides the basis for understanding root causes and making appropriate process improvements. If the culture of the team or organization considers defects as negative, people spend more time trying to avoid defects and also are less likely to report a defect when encountered. This can lead to some defects being identified later in the process, when they are harder and more costly to fix.

Based on my experience, organizations which consider defects as part of the process seem to be able to deliver high quality software faster than organizations which consider defects negative events for which blame should be assigned.

---

Krishen Kota is a veteran of the information technology consulting industry and is President of AdminiTrack, Inc. ([www.adminitrack.com](http://www.adminitrack.com)), which provides a web-based defect and issue tracking application designed specifically for professional project or business teams. Krishen may be contacted via Linked-In at [www.linkedin.com/in/krishenkota](http://www.linkedin.com/in/krishenkota) or via Twitter at <http://twitter.com/KrishenKota> .