

The Importance of Defect Tracking in Software Development

By Dan Suceava, Program Manager, Axosoft LLC
dans@axosoft.com

THE SOFTWARE DEVELOPMENT CYCLE	1
DEFECTS – WHERE DO THEY COME FROM?	3
BUGS – DON'T LET THEM GET AWAY	4
ONTIME DEFECT MANAGEMENT SYSTEM	7

Software development can be a very complex and overwhelming task that requires good organization and management. This document's objective is to show how important one certain aspect of management - defect tracking - is to a successful software development process.

The Software Development Cycle

Most software development teams follow, to a certain degree, a software development cycle to ensure successful completion of their project, be it a product they sell, a custom application, or an internal tool used by the company. Organization and structure is key to success, and teams that don't have either will likely fail.

A software development cycle consists of stages that guide the development process and define the steps necessary for successful completion of the project. These stages don't necessarily define an ordered checklist to be followed, they can be started and worked on simultaneously, and most likely they will be repeated multiple times during the cycle.

Development cycles include, but are not limited to, the following stages:

Requirements Gathering

This stage consists of the collection of requirements for the software application. The requirements describe what the application should do, how it should work, how it will be used, and any other information that helps in the

Software development cycles include some or all of these stages:

- **Requirements Gathering**
- **Architectural Design**
- **Development**
- **Testing**
- **Deployment**
- **Maintenance**

development of the application. The requirements are basically the customer's needs, and are usually collected directly from the customer through a process of questioning and interviewing.

Architectural Design

The design stage involves, at the very least, a high level design of the system. A design document should outline all the major components of the application, preferably even all the smaller components, the functionality of these components, and the relationship between all the components. The design should detail exactly how the application will be built without doing the actual coding.

Development

This stage is where the actual coding of the application is done. When most people think of software development, they mainly identify with this stage, and they think that this is all that is involved in developing an application. When following a good, structured development cycle, the total time spent in this stage compared with the total of all stages in the cycle should be minimal.

Testing

The testing stage involves the testing of the application and the identification of problems, it is quality assurance. Testing can be, and should be, a very complex process through which it is determined whether the application meets the needs of the customer, operates as expected, and doesn't have any, or very few, flaws in it. Testing is the one stage that can span the entire development cycle, and you can never do enough of.

Deployment

This is the stage where the application is ready for the customer, whoever the customer may be. It is ready to be released. It needs to be installed on the customer's computers and it needs to be set up for them to use it. Enterprise-style applications usually require custom implementation and training to get the customer up and running with the application.

Maintenance

The product has been released, customers have purchased it and are using it, and everyone thinks the development cycle is over. Not quite. This is the last, and probably longest of the stages. This stage is where you have to maintain and support the product. Customers will always find problems, big or small, and they will want them fixed. And customers will always want the application to do more, they want more features and additions. The cycle must go on.

Defects can occur at any stage in the software development cycle due mainly to human error.

Defects – Where Do They Come From?

In a perfect world, with perfect people, software development would be a very simple process. We would go through each stage of the development cycle once, and we would be done. As a matter of fact, we would not need to test at all, or maintain the application, we are, after all, perfect.

In the real world we have to deal with not-so-perfect software applications, applications with ... defects. But what are defects? And where do they come from?

Anything that is wrong with the application, whether it is something that causes the application to crash or just a minor spelling mistake, is considered a defect. When the application does not meet the requirements, or it does not perform as expected by the customer, it has defects. When the application does not look good to the customer, or it does not 'feel' right, it has defects.

Defects are not always coding mistakes done by the developers. Most defects are, but not all. Defects can come from any stage in the software development cycle, because every stage involves people.

In the requirements stage, when communicating with the customer, there is plenty of room for error. The customer's needs may be misunderstood or misinterpreted, the customer may miscommunicate his or her needs, and even the requirements document writer may inadvertently cause errors.

A defect is anything that is wrong with the application that causes it to not meet the customer's requirements.

In the architectural design stage, an inexperienced or careless designer can always cause problems later on. Making the wrong choices for the architecture of the application or certain components, designing the wrong functionality, or lack of functionality, and overlooking component relationship and interfacing can spell trouble later in the development process.

The development stage is probably responsible for most of the defects in all applications. This is where human developers 'communicate' with the computer. They have to translate the design document into code, they have to come up with algorithms, and even the simple task of typing on the computer could cause errors. And these usually include the hardest defects to find and fix.

The testing stage is where most defects are found, but some defects can be introduced here as well. Testers usually take the requirements and write test cases against them, create testing scripts, and use automated testing software. Any of these actions can cause errors in the testing process. These may not necessarily cause defects in the application itself, but they certainly affect the development cycle of the application.

Maintenance may include any or all of the above stages, since it involves anything from fixing defects found by customers to adding new features to the application. Thus the cycle repeats itself.

Bugs – Don't Let Them Get Away

Hopefully by now you are convinced that defects do occur during the software development cycle, and they can occur at almost any stage. Even if you don't follow rigidly some kind of software development cycle, and you just do ad-hoc development, defects will occur. In that case you will actually most likely see even more defects because of the lack of structure and organization.

The question you should be asking now is "What can I do about it?" Prevention is the main thing, but while there are many things that can be done to prevent a lot of the defects from being introduced into the system in the first place, the fact is that errors will still be made.

It is important to record and track all defects during the software development process so they can get addressed and fixed on time.

The most important thing left to do, then, is to make sure that all these defects get addressed and fixed, and that they get fixed on time, and retested, so the project schedule doesn't slip. Therefore it is imperative that all defects get recorded and tracked so that the team can be alerted of any issues and they can properly validate and address them.

A good defect management system is, therefore, a necessity for all software development teams.

Record Defects

A team must be able to, first and foremost, record all aspects of what constitutes a defect. Any and all information about what, how, when, where the defect occurred should be entered in the system, in whatever format is necessary, including any screenshots, documents, or any other attachments that help in the validation and fixing of the defect.

Most teams have their own way of tracking defects, and they all have special things that they want to track. The defect management system must allow for a wide range of customization to provide the team with a way of tracking exactly what they need.

Notify the Team

Most times when defects are found they are not found by the same person who caused them or will be fixing them. In large teams, with a complex software development process, defects have to go through many people to ensure they get properly fixed. When these defects are entered into the system, or when they are handed over to other people, the people involved need to know about it right away so they can address them in a timely manner. The defect management system should be able to automatically notify the appropriate people, and should allow the people entering the defects into the system to notify anyone they deem necessary to know about the defects. Proper communication is key to quick and timely response from the team.

Let It Flow

A lot of teams have a process set up through which they ensure that defects are properly verified, validated, and fixed.

For example, when a defect is entered into the system it may not necessarily go directly to a person responsible for fixing it. First, it may need to be verified to make sure it is reproducible. Once it's reproduced someone else may need to validate that it is indeed a defect and not something else (like user error, or a feature). Only after it's been validated is the defect handed over to the person responsible for fixing it. From here it is handed over to someone else for testing.

Different teams have different workflows through which defects go. The defect management system must allow these teams to easily set up such workflows that make, among other things, handing over of defects from one person to another very simple.

Organize and Secure

Managing multiple development projects can be a daunting task, and organization is key to success. Being able to keep projects separate, and divide big projects into subprojects can make a manager's life much easier. A defect management system should allow tracking of defects by project so that different projects can be tracked and managed independently of one another.

Having multiple projects often means having multiple development teams, each working on their own project. Being able to secure the system so that individual team members can only access the projects and items that they are working on is important to keeping a development team on track and focused.

Manage for Success

To lead a successful development team, and to complete software projects on time, project managers need to always know what the status of the project is, what the team members are working on, and how much work there is to be done.

A good defect management system can help project managers achieve successful product development.

A good defect management system can help managers achieve this success. It can keep them current with the project's activity by notifying them of all the latest defects entered in the system. It can let them browse defects in the system and easily search and find specific defects. It can print reports that identify the status of the project, the amount of work done or amount of work left.

OnTime Defect Management System

A good defect management system can really make a difference in the software development cycle. It can be the difference between developing a good or a great software application. Axosoft's OnTime 2005 is one such defect management system.

OnTime 2005 tracks extensive standard defect information such as name, description, steps to reproduce, resolution, notes, project, status, priority, severity, date found, date fixed, who reported the defect, who the defect is assigned to, estimated duration, actual duration. If the standard information is not enough, OnTime allows you to create an unlimited number of custom fields that can track any other information required. If that is too much, OnTime allows you to select the fields you want the users of the system to be able to enter in.

OnTime 2005 can be set up to notify any person, through e-mail, any time a defect is entered into the system, modified, or deleted. This can be set up globally for the entire system, at the project level, or at the defect level. At each level you can set up a list of people to get notified on specific events. The list can contain users from the system, or any other e-mail.

OnTime 2005 provides an easy to set up and use workflow system for defects. The workflow consists of steps that a defect can be put through. A workflow step can be set up to set the defect's status, priority, and severity, re-assign the defect to another user, and force editing of certain fields. It can also be set up to send a notification e-mail to certain people when the defect is put in that workflow step. Any number of workflow steps can be defined in the system.

Axosoft's OnTime Defect Management System is the solution for software development teams looking to improve their development process.

OnTime 2005 provides for hierarchical definition of any number of projects and subprojects. This allows for simple organization and management of defects by project. OnTime's role-based security model can be set up to restrict users from accessing projects and defects in the system that they are not supposed to be working on. Users can be set up to have different privileges on different projects.

OnTime 2005's powerful and easy to use UI can make listing and searching for defects very simple. Looking for defects in a specific project is as simple as selecting the project from the list. Searching for defects by any number of fields is fast and easy to use. OnTime's powerful filters can be set up to filter defects in the list by any criteria based on the defect's fields. OnTime can then export or print a number of reports based on the defects in the list. And the powerful project reports can give managers a quick view of the status of the project.

Axosoft's OnTime Defect Management System is the solution for software development teams looking to improve their development process.