

CompanyName Engineering

Quality Assurance

Quality Module Defect Control Process

Document Version: V1.8

Date: 3/4/2000

DRAFT

CompanyName Engineering

Quality Assurance

SIGN-OFF SHEET

The following approvals are required to initiate the Defect Control System process:

Approval Name:	Department	Approval Signature:	Date Approved:
	Engineering		
	Operations		
	NetOps		
	Technical Support		
	Project Management		
	Engineering – Quality Assurance		
	Engineering – Development		
	Engineering – Development		
	Engineering – Development		
	Engineering		

The above sign-offs are individuals that are intimately involved in the successful execution of the defect control process.

Other Project Participants for Distribution Only:

Quality Assurance Team
Development Team
NetOps Team
Project Managers
Technical Support

SIGN-OFF SHEET.....	II
PREFACE.....	IV
PURPOSE.....	IV
AUDIENCE.....	IV
DOCUMENT HISTORY.....	IV
1.0 PROJECT DESCRIPTION.....	1
2.0 DELIVERABLE ITEMS.....	1
2.1 DOCUMENTS.....	1
3.0 GENERAL SYSTEM AND PROCESS OVERVIEW.....	2
3.1 EMERGENCY PROBLEMS.....	2
3.2 DEFECTS WITH WORK SEQUENCE NUMBERS.....	3
3.3 PROJECT OR RELEASE DEFECTS.....	4
3.4 DATA MODIFICATIONS.....	4
4.0 ROLES AND RESPONSIBILITIES.....	2
4.1 ORIGINATOR.....	2
4.2 COMMITTING A RECORD.....	3
4.3 DEVELOPMENT TEAM.....	4
4.4 QUALITY ASSURANCE TEAM.....	4
4.5 TECHNICAL SUPPORT TEAM.....	5
4.6 CONFIGURATION MANAGEMENT TEAM.....	5
4.7 NETOPS TEAM.....	5
4.8 PROJECT MANAGERS TEAM.....	6
5.0 COMPANYNAME – QUALITY MODULE ADMINISTRATION.....	6
6.0 REPORTS AND METRICS.....	6
7.0 DEFECT STATUS OR STATE.....	7
8.0 EXECUTIVE STAFF REPORTS (EREPORT).....	9
APPENDIX A – PROCESS DIAGRAMS.....	10
APPENDIX B – DEFECT DEFINITIONS – SEVERITY/PRIORITY.....	12
<i>PRIORITY DEFINITIONS.....</i>	<i>12</i>
<i>SEVERITY DEFINITION.....</i>	<i>13</i>

Preface

This is the Defect Control Process document that will describe the processes used for managing defects and other documented areas, in the CompanyName Quality module.

Purpose

Audience

This plan is intended for use by CompanyName Team Members responsible for the using the CompanyName Quality module.

Document History

Version	Date	Who	Revision
V1.0	1/7/200	FAL	Created
V1.1	1/11/00	FAL	Added update to section 4.1. Added new sections 6.0 and 7.0
V1.2	1/11/00	FAL	Updated section 4.1 to include types and assignments to TS.
V1.3	1/11/00	FAL	Updated 4.1 Originator process and added section 8.0 Executive Staff Reports
V1.3	1/11/00	FAL	Update Next In Line process section 3.3
V1.4	1/17/00	FAL	Update 4.1- New fields Assigned To and Date Assigned, defect Type Next In Line included for all new defects to be filtered (not project or release related). Section 7.0 updated for Status = Verified Production Ready.
V1.5	1/20/00	FAL	Added Section 4.7 – Project Managers Team
V1.6	1/26/00	FAL	Changed Section 3.5 to change process for Data Modifications and section 4.1 to delete TYPE Data Mod Code change.
V1.7	1/27/00	FAL	Added 4.2 Committing A Record
V1.8	3/4/00	FAL	Appendix A changed: Severity and Priority. Next In Line, Top5 were deleted as types. Showstopper no longer exists for Severity and Priority, already there.

1.0 Project Description

2.0 Deliverable Items

The following documents will be written and placed under Configuration Management and Document Control.

2.1 Documents

- Defect Control System Process

3.0 General System and Process Overview

The CompanyName Quality module is the system for documenting the progress of the software product development process. It can be used to measure and track the quality of the application as it matures and evolves through the development life cycle.

The success of this tool is dependent upon the members of the team and its dedication to following the defect control process. Collecting accurate metrics and moving on to the next step, in resolving problem areas, can only be achieved through team members communicating through the use of the CompanyName Quality module.

Each group, within each department, is dependent on other groups to follow the correct Quality module procedures. These correct procedures are defined in this document.

Each group or team member has a responsibility to enter the required fields requested or the entire Quality module will not be successful, in establishing control over the product development process. Required fields for each group or team member are defined in this document.

The Notification (automatic emails to the next Owner of the problem (BIC)) process is completely driven by the changing of specific fields and the most important field is the Owner field. Owner field is similar to Ball In Court (BIC). If you do not change the Owner field to the correct person or do not change the Owner, then you will still own the problem. Automatic emails are sent to the next Owner, when the item is completed along the process. Other automatic emails are sent to the creator of the problem area and to interested parties involved with the problem.

Specific queries for collecting metrics or specific project information are available to everyone who has access to the Quality module. If fields are not updated correctly, then these metrics and project information will be misinterpreted.

The two individuals for maintaining the correct flow of this process are the Manager of Technical Support and the Director of Quality Assurance.

3.1 Emergency Problems

An emergency problem means a customer is down and can not get any work done, because of the reported problem. An emergency problem is also related to daily emergency Data Modifications that need to be made, to keep the customer up and running. This kind of problem is a Priority 1 (P1) and will be designated as a Priority 1 in the reported defect, with a Severity of S1 and a Priority of P1. An emergency problem is not who yells the loudest or the customer threatens to quit

using the application. There are occasions when exceptions to what is an Emergency fix will be exercised and these exceptions will be at the discretion of the Technical Support Manager, Marketing and the Director of QA.

The Technical Support Manager, or his designee, is the main filter for determining what is an emergency problem. The Technical Support person does not make assignments, and will not be addressed by development, unless the problem is an emergency or assigned as a defect with a Work Sequence number. That decision is filtered by the Technical Support Manager and works with the Director of Quality Assurance to enforce this process. The Technical Support Manager, with Marketing also assigns a Work Sequence number to the defect, so that development knows what to work on next.

Emergency day-to-day Data Modifications, without code changes, are returned fixed to the creator (Owner), in Technical Support (Open and Ready for Technical Support), for testing. Emergency fixes and Data Modifications with code changes are tested by Quality Assurance (Open and Ready For QA). Emergency fixes and Data Modifications with code changes are sent to the QA Group (Owner), for assignment to a QA Engineer. There may be high peak periods where the Director Of Quality Assurance will request assistance from the Technical Support Manager, to assist in testing areas that are normally not tested by Technical Support.

3.2 Defects with Work Sequence Numbers

Defects that are filtered are given Work Sequence numbers and Development will work on these defects, in the order received. As resources are available some of these defects may be grouped and fixed by a number of different developers and therefore some fixes may not necessarily be in the order received.

Some of these problems are very serious problems and must be pushed by the creator, to champion that the fix gets done immediately. These problems are filtered through the Technical Support Manager and are given Work Sequence numbers. When a problem is fixed by development, the Owner is changed to the QA Group. These defects are tested by QA Engineers, but assistance may be requested from the Technical Support Team. When completed with testing, automatic Notifications will be sent to the appropriate individuals for further processing.

3.3 Project or Release Defects

Defects found, during a release cycle, or project are handled by the Quality Assurance Team. Defects fixed by development are returned to the QA Engineer that created the defect. Change Owner to QA Engineer that created the original defect. All of these defects will be assigned a Target Release number related to the project.

3.4 Data Modifications

Data modifications are modifications to data or information that the customer is using and the CompanyName application does not allow the customer to make the modification. Typically, these problems are considered data mod emergencies and must be fixed immediately. Data modifications are tested by Technical Support and are immediately put into production.

The Data Modifications that QA tests are Data Modifications that require a build and will have to be tested in QA2, before releasing into production. The TYPE for this will still be Data Modification. However, the assignment is to QA GROUP, after the code is place in QA2. Developer assigned, fills in field Migration Path to QA2. Technical Support only tests Data Modifications that do not require a build and are going directly into production.

If a problem report is initially defined as a Data Modification and it is a Data Modification, then Technical Support handles it. However, if the Data Modification requires a code change, then it is no longer a Data Modification. The Data Modification TYPE now becomes Defect. The assigned developer or Development Manager changes TYPE from Data Modification to Defect and OWNER to TS GROUP. The TS GROUP Manager will review the defect and give it a Work Sequence number.

Data Modifications, Emergency, and other defects with Work Sequence numbers are problems currently happening in production. All problems, regardless of the importance, may not get fixed within the expected time frame, because some problems are more complex than others. The important thing to remember is to communicate, communicate, communicate, with development, with QA and most importantly with the customer. Get the details of the problem and why it will take so long to fix and then work with your customer, to keep them informed.

Sometimes these problems may turn into an RFE (Request For Enhancement). The Marketing Team and the Product Council handle RFE's.

4.0 Roles and Responsibilities

4.1 Originator

The first thing the Originator should do is to do a query on the Abstract or Extended Description with key words to determine if the defect already exists or previously existed. This will allow us to keep track of reoccurring problems that were not previously reproducible and avoids duplicate defects.

The next thing the Originator should do is, if this is a new defect and it needs to be filtered to determine if it is an Emergency, defect with a Work Sequence number, RFE or it gets assigned a low Priority and it does not make it into any list (sits in the data base, until someone decides it needs to be part of a future release), then **Status** should = **Open**, **Sub Status** = **Unassigned**, **Owner** = **TS Group** and **Type** = **Emergency, Defect or RFE**. The TS Manager will filter out these defects and decide where it should go and give it a Work Sequence number. Defects going into a Release are handled by QA. In the future, a field called **Engineer** will be added, so that you will always know who the developer is on a defect. **Engineer** is filled out by the Developer working on the problem and will also fill out **Date Assigned** (future field), when work is actually started. **Owner** is a **BIC** field and always changes.

Types:

1. **Training:** This is a training defect and should be resolved through training.
2. **Design Flaw:** This is really not a defect, but a problem with the design.
3. **RFE:** This is a Request For Enhancement for ALL other CompanyName products or applications. Use field Product to distinguish an RFE from other products.
4. **Defect:** This is a defect and does not function as designed. Customer document problems are considered software defects.
5. **Y2K Issues:** These are defects that are year 2000 date related.
6. **3rd Party Incompatibility:** The CompanyName application is having a problem with a 3rd party application (CompanyName Viewer, Whip, Reader).
7. **Data Modification:** This is a data problem that a customer is having a problem with and it is stopping work progress. When the Data Modification is fixed, then this Data Modification is sent to the originator in TS. TS will test these.
8. **User Documentation:** This problem may require a document change to User Documentation, not Customer Doc ID. Customer Doc ID is a Type = Defect, because it is a software problem and use Doc ID on defect form.
9. **Hardware Issue:** This is obvious.
10. **Emergency:** Already explained.

The Originator creates the defect. Required fields to fill out are: **Abstract** – Summary of the problem. **Extended Description** – Detailed description of the problem. **Type** – the type of problem found. **Severity** – How serious is the problem. **Priority** – What is the priority of this fix. **Product or Feature** – What is the CompanyName feature or function related to this problem. **Server Name** – What server environment was the problem found. **Target Release** – This could be a fix for Production, CompanyName V1.0, CompanyName V2.0, or some specific release or project. **Status** – Open. **Sub Status** – Unassigned. **Sub-Status – Assigned** can only be done by the TS Manager (or designee), TS Engr (Data Mods only), QA Engr (For Releases, Projects only or special testing requirements), or Development Manager. **Owner (BIC)** – What person owns the resolution to this problem. This person will be automatically notified of the problem. If more information is required **Attachments** can be created and sent with the defect. If **Steps To Reproduce** are required, then use Solution Search and pick Steps To Reproduce for further information that can be used to resolve the problem.

Activities – May be used by everyone to document any outstanding issues that may be blocking progress for getting this defect or problem resolved. If something is blocking progress, document it and make sure that someone owns solving the problem (a name) and that you get a date, as to when the issue it going to be resolved. This is the reason for the Activities section.

4.2 **Committing A Record**

When you make changes to a Product Defect you should be in one of the following screens, to commit the record:

My Product Defects
My Team's Product Defects
All Product Defects

After you update the record, make sure that you commit the record by clicking your mouse on one of the above screens.

Do not make changes to when you are on one of the following screens, because the record will not get committed:

Activities
Attachments
Explorer
Product Defects
Solution Search
Service Requests
Audit Trail
Charts

Some of you may or may not have some of the above screens. The screens you get are based on your Position Assignment.

In the future, the screens: Activities, Attachments, Explorer, Product Defects, Solution Search, Service Requests, Audit Trail and Charts will be read only and you will not be able to make changes to the Product Defect, but you will still be able to do Attachments, write in Activities, do Solution Search, etc.

4.3 Development Team

Development Team members receive a defect assigned to them. Developers will only react to problems that are assigned to them as Data Modifications, Emergency, Top 5, Project/Release related or an RFE assigned to a Release. No other problems will be addressed. If the problem does not fit into any of these categories, it will not get fixed. They will always get notified of problems assigned to them, but should only work on problems that are part of the official category list. If there are any exceptions, the Development Manager or Team Lead should be notified.

After a fix is completed, the required fields to fill out are: **Status** – Remains Open. **Sub Status** – Changes to Ready For TS (Data Mod) or Ready for QA (Data Mod with code changes, Emergency, or Defects). **Migration Path** – Where was the code placed for testing. **Owner** – TS originator (If Data Mod), QA Group (If Data Mod with code change, Emergency or Defects), QA originator (If Project/Release related). **Resolution** – What was the resolution to the problem.

The **Activities** section is used to keep notes on the progress of the defect and to keep other informed of what is happening with this defect. Developers should live and breathe this section.

4.4 Quality Assurance Team

After a defect is tested the required fields to fill out are: **Status** – Changes to Closed. **Sub Status** – Changes to Verified Production Ready (unless it fails, then it is sent back to developer). **Owner** – Changes to CM Group, to inform that the defect is now ready to be built.

If fix fails, then: **Status** – Remains Open; **Sub-Status** – Changes to Fix Failed. **Rework** – Changes to 1 sequentially. **Owner/BIC** – Changes to original developer.

Also use **Attachments** and **Activities** where necessary, as stated under Originator.

4.5 Technical Support Team

After a Data Mod is tested, the fields to fill out are: **Status** – Closed. **Sub Status** – In Production. If it fails send it back to the original developer (**Owner/BIC**), with **Status** – Open and **Sub Status** – Fix Failed and Rework – Changes to 1 sequentially.

If Technical Supports is working with QA to assist with testing, then they should follow the same rules as section 4.4 and act like QA Engineers.

Also use **Attachments** and **Activities** where necessary, as stated under Originator.

4.6 Configuration Management Team

The CM folks are responsible for working with the Technical Support Manager to determine what will be built, based on all the defects with a **Status** of Closed and **Sub Status** of Verified Production Ready. They are responsible for generating the **Release Notes** for the NetOps Team. If items are **Emergency or defect for Production**, then these items are candidates to be built. If items are assigned to a current **Project/Release**, then these items are held in limbo, until the Feature of the related Release is ready to be shipped and put into production. In this case, **Release Notes** are written to include all the items to be built for the Release. If appropriate, **Build Numbers** should be associated with each defect fixed, so that we will know when the defect was actually shipped out and on what build.

Also use **Attachments** and **Activities** where necessary, as stated under Originator.

The CM person changes **Sub Status** to Ready for NetOps when the incident is ready to be moved into Production.

4.7 NetOps Team

NetOps is responsible for code moving into production (Stage, Demo and Production). The migration of code should normally be Stage, Demo and then Production to minimize the impact of problems as code migrates. All performance testing (if required) should occur on Stage, before code is migrated into Demo or Production.

NetOps is the gatekeeper into the Stage, Demo and Production environments. They will be responsible for making sure that CompanyName is updated with **Status** – Remains Closed and **Sub Status** – Changes to In Production. **Code Migration** – Changes to In Production. This will give us a history (Audit Trail) of the code migration, as it moves from server to server. It is very important that the

Sub Status field changes to In Production, otherwise we will not know what has to be built for the following week, if everything remains **Sub Status** – Ready For NetOps.

Also use **Attachments** and **Activities** where necessary, as stated under Originator.

4.8 Project Managers Team

PMs only need to understand 2 processes. When creating a CompanyName **Defect**. **Type** = Defect, **Status** = Open, **Sub-Status** = unassigned and **Owner** = TS Group.

For a **Data Modification**: **Type** = Data Mod, **Status** = Open, **Sub-Status** = Unassigned and **Owner** = TS Group. Owner is like BIC. Refer to Section 4.1 (Defect Control Process) for additional inputs.

5.0 CompanyName – Quality Module Administration

The administration of the Quality module is the responsibility of the Director of Quality Assurance. The following are the responsibilities of specific individuals:

Field Values: Director of QA or QA designee

Queries and Filters – Universal (Director of QA or QA designee) and Local (Anybody). You can create queries for Universal use, if you think they are useful to everyone, but the Director of QA will release the query for Universal use.

Build Numbers: The Build Numbers are entered by the CM persons to keep track of what build a particular defect is associated with.

New Product Names: For the Quality Module ONLY – Director of QA (Or QA Designee)

New Target Releases: Same as above.

User Ids/Passwords and Roles: IT designee

6.0 Reports and Metrics

The following reports and the definition of the reports are universal queries available for your enjoyment:

1. **Report – Emergency (Open)** – This report will show all Emergency open items currently assigned to development and are actively being worked on.
2. **Report – Master List – Open By Work Sequence** – This report shows all the work that has been filtered by the Technical Support Manager and has been assigned Work Sequence numbers for development to work from.
3. **RFE Release – In Engineering** – These are all the RFE open items that Marketing would like to see Engineering work on along with the current

release. Marketing owns this list and works with Engineering for scheduling work.

4. **Release X.X** – This report will contain the most current Release, being worked on by Engineering, of all open items for the entire release.
5. **Product** – *product name*. This report will contain all of the open items for a specific product. You can also create these simple queries yourself by doing a Query, putting in the product name in the Product column and then choose Open in the Status column. Execute the query and if you like it, save it and it becomes your local query.

7.0 Defect Status or State

The following queries allow you to get status on what current state is a specific defect. So, if you want to know did the defect make it into production or where is it in the product life cycle, this is how you do it. Normally, the defects tracked here, on a real-time daily basis, are Data Modifications, Emergency, Top 5, related to the current Release and selected RFE's. If the defect is not in any of these queries, then Engineering is not assigned to work on the item. See the TS Manager for Emergency, Data Modifications and Top 5. See the Director of QA for the current Release and see Marketing for RFE's.

1. **Status - Ready For TS:** This means that there is a Data Modification that needs to be tested by a Technical Support person. If the defect is here, it is scheduled to be tested and it is not in production yet. When the TS person finishes testing this, then he will move the defect to Closed – In Production.
2. **Status - Ready For QA:** This means that there is an Emergency, Top 5 or Release defect fixed and ready to be tested by QA. When QA is done testing the defect is moved to Closed – Verified Production Ready. The Owner is also changed to CM Group.
3. **Status – Verified Production Ready:** This means that the fix was tested and is now ready to be built and put into Stage, Demo, Sneak Peak and/or Production. The CM person (Configuration Management) then creates Release Notes for NetOps and moves the defect to Closed – Ready For NetOps. There is Verified Production Ready for a specific product or items for Production, so that builds do not contain items targeted for a release or product not ready for production. Make sure you are using or create a query/filter to get the right items you want to build.
4. **Status – Ready For NetOps:** This means that the defect is now in NetOps's court to make the build and get the items into the appropriate environment (Stage, Demo, Sneak Peak and/or Production). After the defect is built, the IT person moves the defect to Closed – In Production.
5. **Status – In Production:** This status will show **ALL** items that are In Production. If you want to know if your item made it into production, then all you do is a query of the defect number and if it shows up, then it's a done deal.

If we all do our part, then the above status will always be correct and additional metrics can be generated. The point of all of this is that you can track the defect you are interested and always find out where it is, in our product development life cycle.

8.0 Executive Staff Reports (eReport)

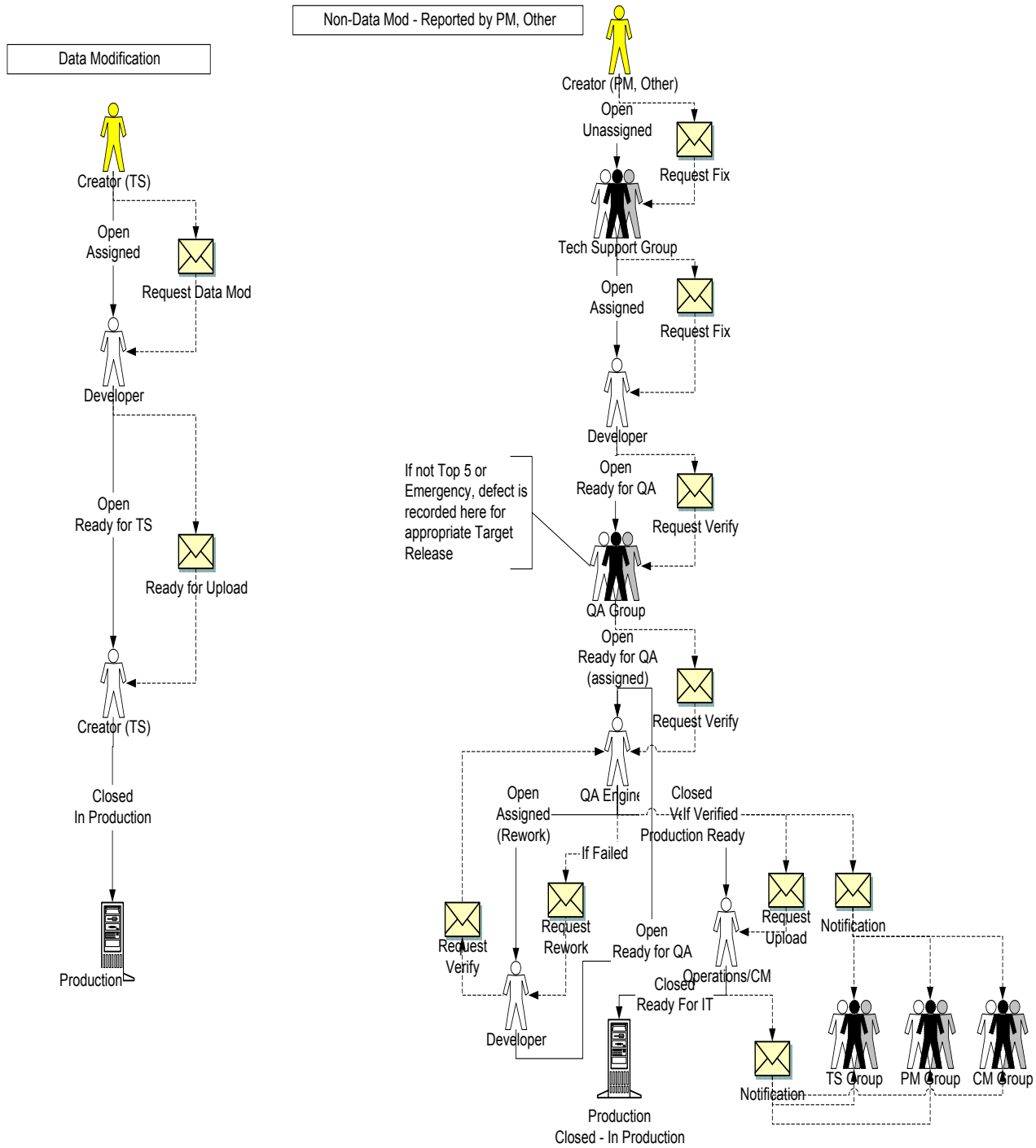
The following reports are provided and updated weekly. The weekly reports cover Monday through Friday and are updated every Friday. These reports were created so that the executive staff can, at any time, view the progress of the CompanyName development process, as it relates to defects and RFE's. Anyone is welcomed to review these reports. The reports are available by moving your mouse to the upper right hand corner of the CompanyName Quality module, and clicking on the down arrow of the label called Queries. Select the eReport you wish to review. To get an accurate count, you must move your mouse to the down arrow slide (scroll button) and press it until it gets to the bottom. Do not slide the scroll button down, because you will not get a count.

1. eReport – Total Open Defects
2. eReport – Total Open RFE's
3. eReport – Total Open P1, P2, P3 and P4
4. eReport – Total Open Data Mods

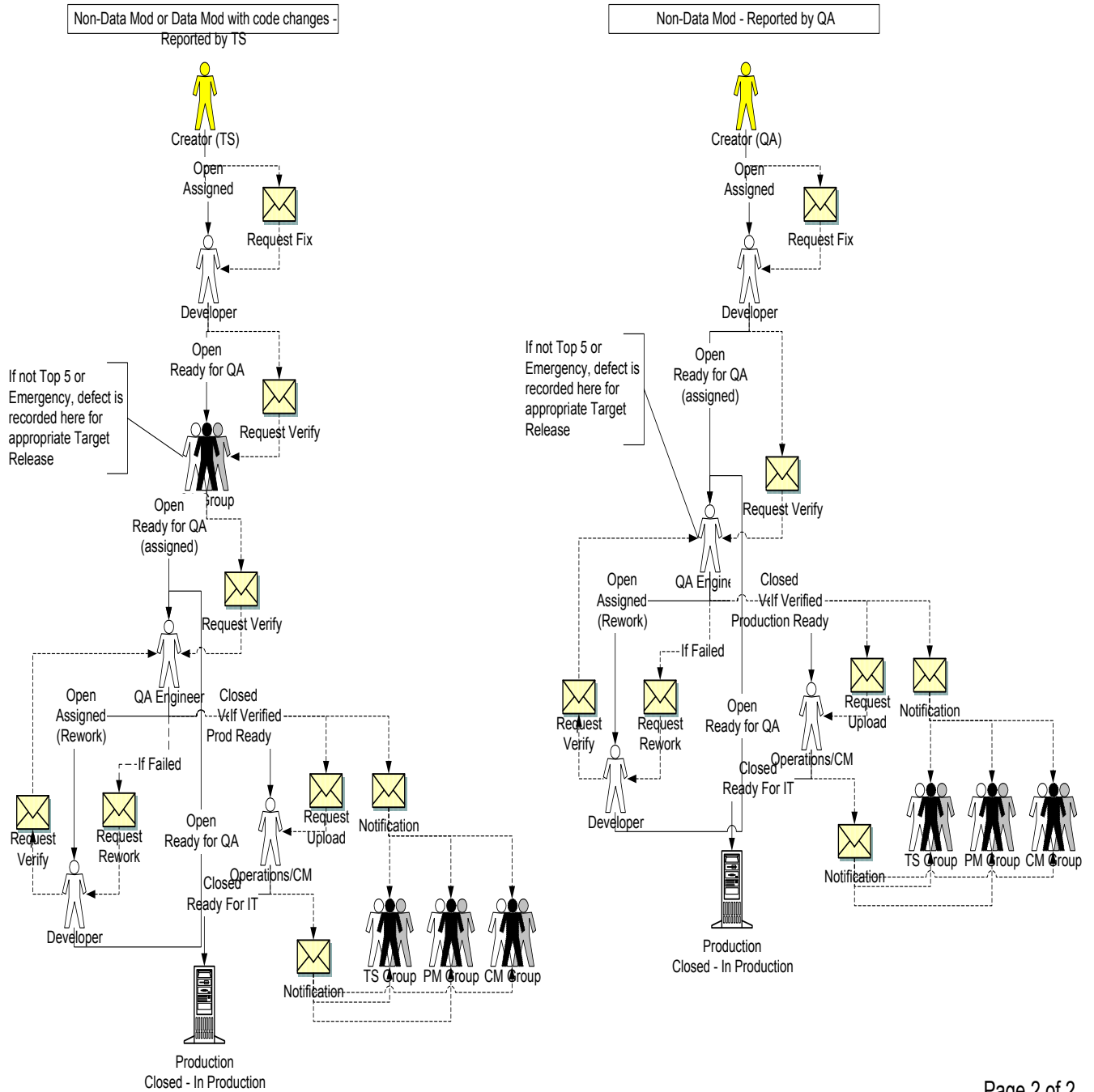
Seagate Crystal Reports will also be used to download data directly from the CompanyName data base and generate reports with metrics, trends, analysis of specific areas and graphs.

APPENDIX A – Process Diagrams

SIEBEL - Quality Defect Tracking Notifications - Workflow Diagram (v2.0)



SIEBEL - Quality Defect Tracking Notifications - Workflow Diagram (v2.0)



APPENDIX B – Defect Definitions – Severity/Priority

A defect is an error in the product, which does not match the code and or the requirements specification, and goes through the formal input, verify, regression, prioritization throughout the development/testing life cycle.

EXAMPLES:

You can have a Priority 1 (P1) with a Severity 3 (S3), because a customer needs or uses that feature in a priority one fashion, but it's severity of the problem may be of severity 3 (S3). You can have a Severity 1 (S1) with a Priority 3 (P3), because the customer may not hit that problem for a long time or may not use it in a P1 definition, but the problem is of S1 definition.

PRIORITY DEFINITIONS

Priority P1: This defect could be a "Mission Critical" defect. It means to stop everything and work on this defect. Criteria should be:

- No Workaround
- Complete stoppage of workflow
- Customer is down
- Data corruption
- System or application crashes
- Loss of an account
- Production is down
- Demo is down
- Performance of Production and Demo is sever enough to warrant a Show Stopper

Priority 2: This defect could be blocking other testing, a functional failure too severe that most or all users or other components of the product are seriously affected, or is all of the preceding with a Severity of rating 1 or 2. **A WORKAROUND IS AVAILABLE.**

Priority 3: This defect is blocking other testing, but is considered a serious problem that must be fixed/resolved before exit of the current phase or release of the product. The Severity of rating can be from 1 to 3.

Priority 4: This defect is considered to be a non-show stopper for shipping the product. Severity rating can be from 1 to 3.

SEVERITY DEFINITION

Severity 1: The severity of this defect could be a "Mission Critical" defect. It means to stop everything and work on this defect. Criteria should be:

1. Customer Down
2. Feature Down
3. Hardware Down
4. CompanyName Application Down
5. System Crash
6. Serious Data Loss – Impacts Customer Workflow

Severity 2: Partial loss of system functionality or partial loss of data. Product will cause some part of the system to crash or hang and the user could experience data loss.

Severity 3: Feature failure that cannot be circumvented. Loss of functionality for which no easy workaround exists.

Severity 4: Feature failure that can be circumvented. Loss of functionality for which an easy workaround exists.