ALM: Executing Test Cases & Defect Management Process

Application lifecycle management Training

IT Delivery Tools Team 2021

Contents

- Describe Objective Evidence
- Execute a test case that passes and execute a test case that fails
- Define a defect
- Identify the three phases of the defect management process
- Compare the four defect severity levels
- Take a defect through the defect management process

Executing Test Cases

- ✓ **Objectives**: Upon completion of this course, you will be able to:

  Describe Objective Evidence

  Execute a test case that passes and execute a test case that fails**.**

- ✓ This course is intended for Test Executors.

- ✓ **What is test evidence?**

  Test evidence is subjective documentation that is collected during testing to support the pass/fail judgement of the Tester on a particular Test case. Test evidence can be interface screen captures or electronic/printed copy outputs from a set of steps.

- ✓ **When must test evidence be collected?**

  There are two instances where test evidence must be collected.

  The first is whenever a test outcome directly verifies that a **requirement** of the system has been met or that requested changes have been completed as expected.

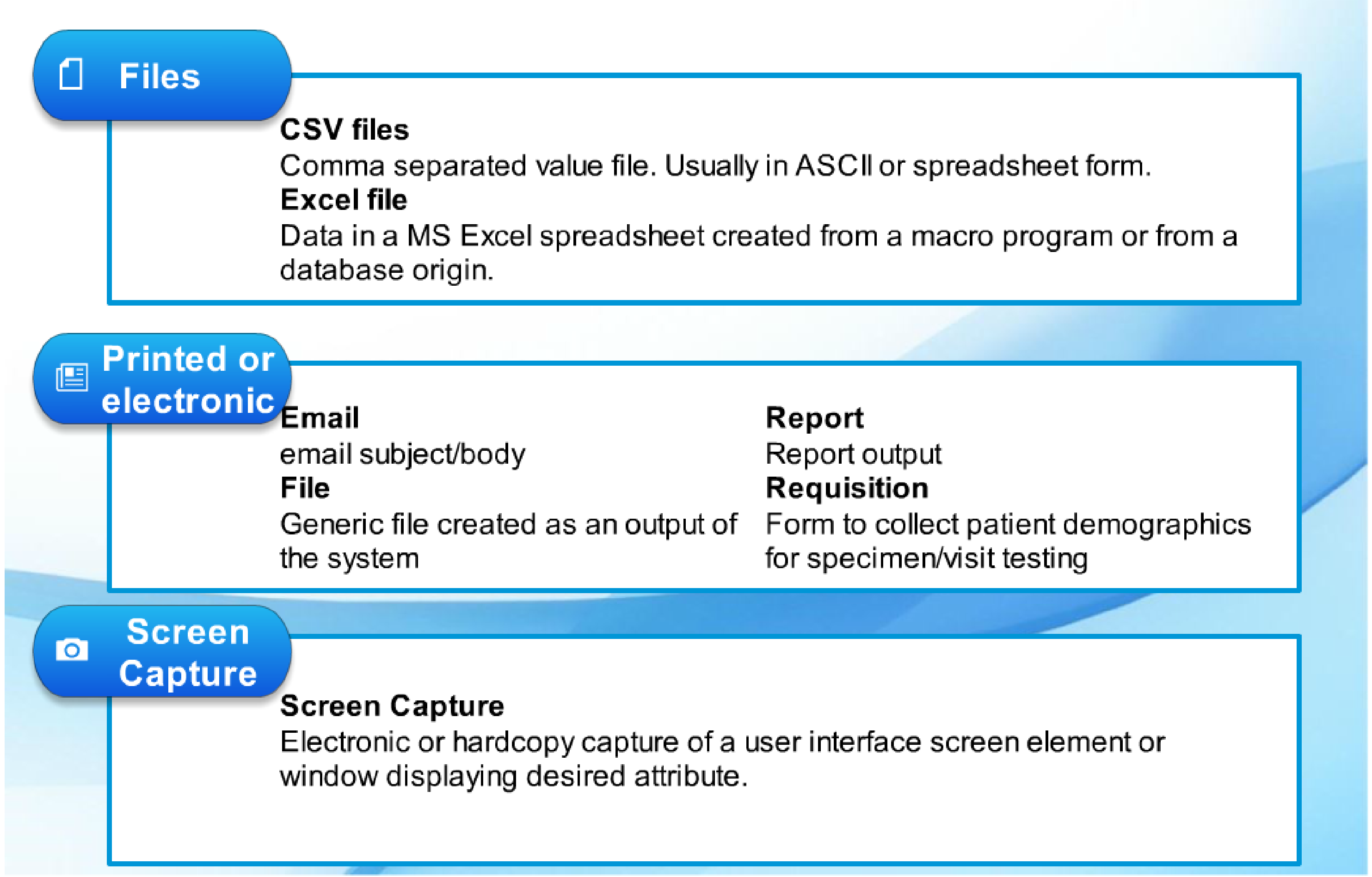
Executing Test Cases

- ✓ The second is when a test outcome is based on verifying a sequence of results, or calculations or other processing operations.

- ✓ **Test Evidence Types**

  There are also three different evidence types.

Executing Test Cases



Files

CSV files
Comma separated value file. Usually in ASCII or spreadsheet form.

Excel file
Data in a MS Excel spreadsheet created from a macro program or from a database origin.

Printed or electronic

Email
email subject/body

File
Generic file created as an output of the system

Report
Report output

Requisition
Form to collect patient demographics for specimen/visit testing

Screen Capture

Screen Capture
Electronic or hardcopy capture of a user interface screen element or window displaying desired attribute.

Executing Test Cases

✓ **Protect sensitive information**

According to IQVIA work Instructions IT225 – SLC Testing, you must protect sensitive information that could potentially breach confidentiality if reviewed during an audit. Patient or sponsor related information are two examples of data that is considered sensitive.

It is the Test Managers responsibility to ensure sensitive data is secured and controlled with the same level or rigor as the productive environment. Prior to submitting any test evidence to an external auditor, all data that could be considered sensitive must be redacted.

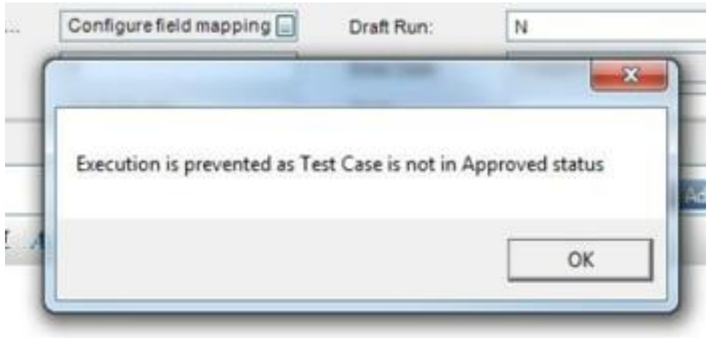
✓ **Execute a Test case**

In this module you will execute a test case. You will be provided steps to accomplish these three tasks: run a test case, add objective evidence, and end a test case.

Any test case that you execute will have already been through the Peer review process and approved by the Test Manager.

If you were to try and execute a test case that has not been approved, you will see this alert window.

Executing Test Cases



Execution is prevented as Test Case is not in Approved status

Task 1: Run a Test Case

Click the **Testing** tab.

Click the **Test Lab** button.

Click the **Execution Grid** tab.

Click **Run**.

Click the **Run with Manual Runner** item**.**

Click the **Build version** drop down button.

Executing Test Cases

Click the **Pre-Build** item.

Click **Begin Run**.

Task 2: Add Objective Evidence

Notice the step requires Objective Evidence. Click the **Attachments** button

The camera icon will capture the screen shot. Click the Snapshot button.

Click and drag the camera to the area of the screen you want to capture.

Click the **Attach** button. Click **Ok**.

Always include the actual result for each step in the test case.

Step 3: End a Test Case

Click the **Pass selected** button

Executing Test Cases

Notice the screen advanced to the next step.

Add the actual result.

Click the **Pass selected** button.

Imagine that the actual result did not match the expected result of Step 3.

Click the **Fail Selected** button.

Enter the Actual result for the failed step test.

When a test step fails, the entire test set fails. You can end the test.

Click the **End Run** button.

✓ In this scenario, you will complete a test case that passes.

Click the **Run** button.

Click the **Run with Manual Runner** item.

Executing Test Cases

Click the **Build Version** drop down button.

Click the **Pre-Build** item.

Click **Begin Run.**

Pass all of the Test steps.

Click the **End Run** button.

Now you can see that the test is passed.

✓ **Restart a Test Case**

What happens if you are in the middle of executing a test case, but you have to work on something else?

You can end the test case, and execute it at a later time, picking up where you left off.

End the case.

You will see that the test case is listed as Not Completed.

Executing Test Cases

When you want to return to the Test Case, select the Continue Manual Run option.

When you run the test case, ALM will return you to where you left off.

✓ **Summary**

You should now be able to:

Describe Objective Evidence

Execute a test case that passes and execute a test case that fails.

Defect Management Process

- ✓ **Objectives**: Upon completion of this course, you will be able to:

  Define a defect.

  Identify the three phases of the defect management process.

  Compare the four defect severity levels.

  Use Application lifecycle management to take a defect through the defect management process.

- ✓ This course is intended for all the users of Application lifecycle management.

- ✓ **Defect Management Process**

  A defect can be encountered at any time during the SDLC. As soon as defect is discovered, it must be created right away.

  IQVIA uses a Defect Management Process to ensure defects found are efficiently triaged and resolved.

  Many roles play a part during the Defect Management Process, but it is the Test Manager’s responsibility that the process is followed.

Defect Management Process

- ✓ **Defect Discovery**

  There are three phases of the Defect Management Process: Defect Discovery, Defect Triage and Defect Resolution.

  Defect Discovery creates a record of the Defect discovered during each test iteration and final validation iteration.

- ✓ **Logging a Defect**

  To accurately create a defect, you need to complete the following required data.

  Defect ID – Unique identifier for the defect (auto populated by Application lifecycle management)

  Summary – A high level summary of the defect.

  Found During – The testing stage in which the defect was found (Pre-validation, UAT, Validation, Production)

  Severity – The severity of the defect relates to its scope/impact on the system.

  Description – Detailed information of the defect. Most of it is populated from the test case step in Application lifecycle management.

Defect Management Process

Detected by – The unique ID of the Tester who raised the defect. This is pre-populated by Application lifecycle management.

Detected In Release – The testing release number during which the defect was detected.

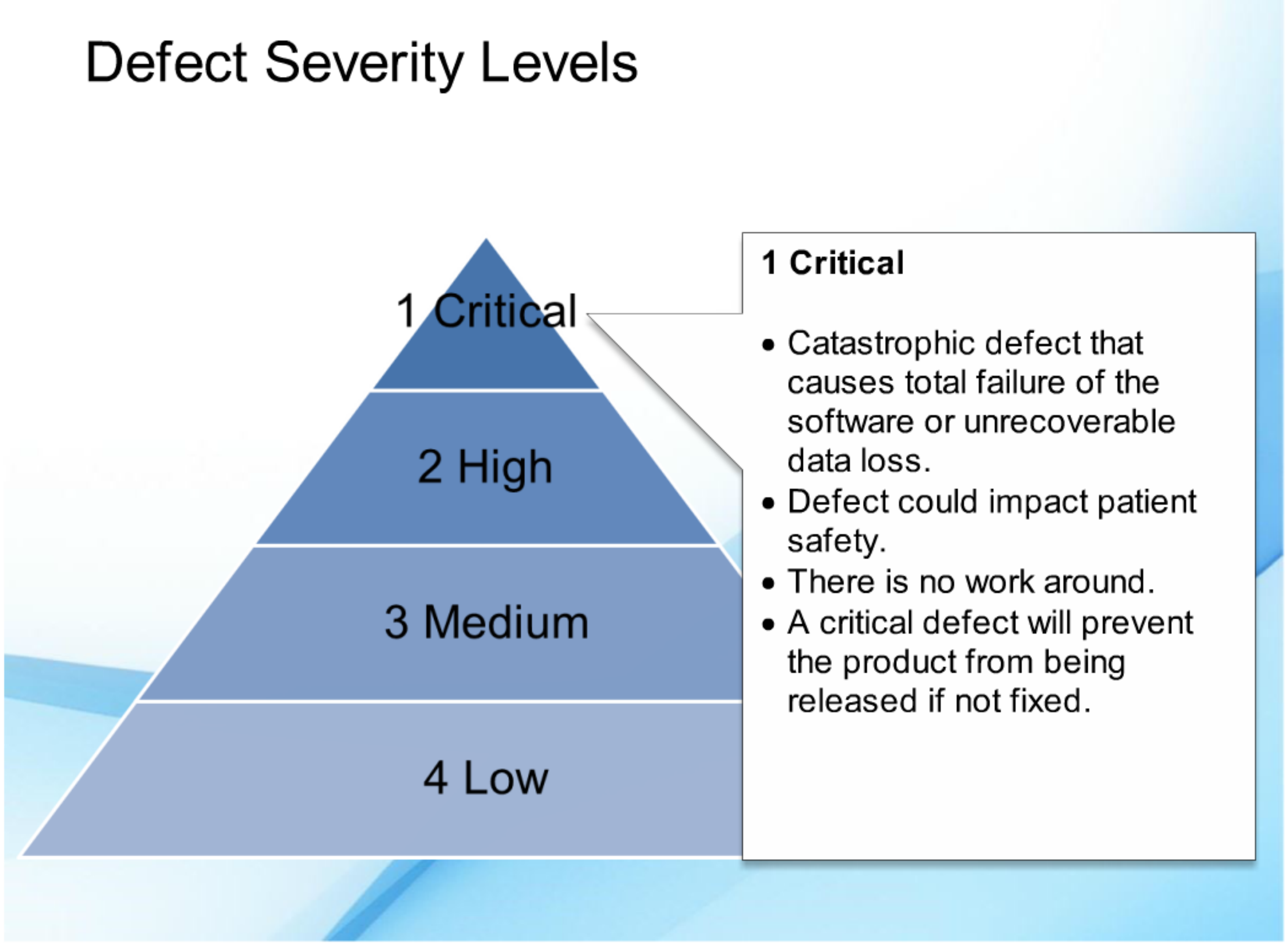
Detected During – The testing iteration during which the defect was raised.

Other defect fields will be required as the defect progresses through the defect management process. For example, once the defect status is set to “fixed” the Defect Root cause field will need to be filled in. Details on mandatory defect fields can be found in work instruction IT225.

✓ **Defect Severity Levels**

The tester opening a defect make a judgement on the level of severity. There are four to choose from.

Defect Management Process



Defect Severity Levels

1 Critical

2 High

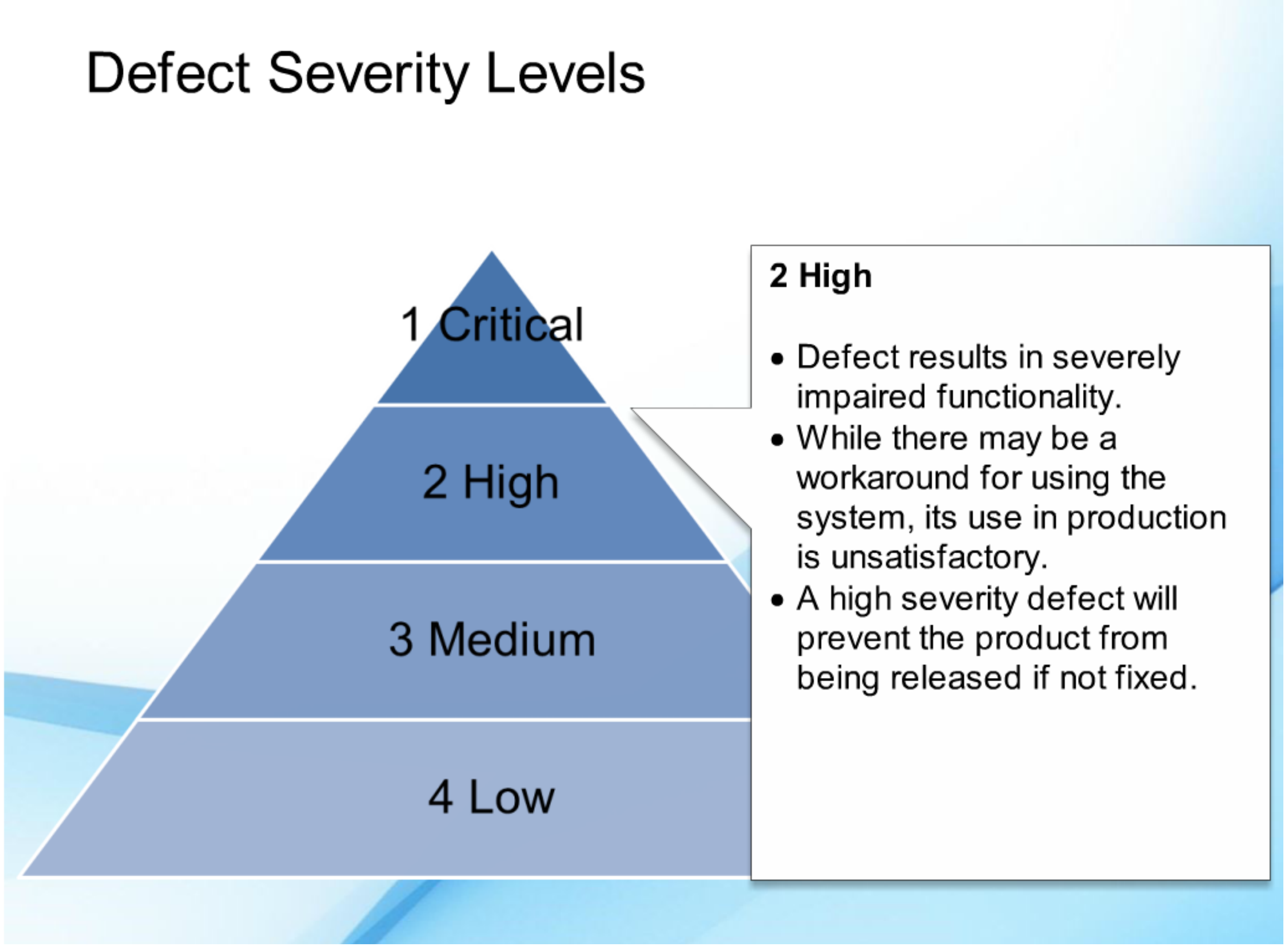
3 Medium

4 Low

1 Critical

- Catastrophic defect that causes total failure of the software or unrecoverable data loss.
- Defect could impact patient safety.
- There is no work around.
- A critical defect will prevent the product from being released if not fixed.

Defect Management Process



Defect Severity Levels

1 Critical

2 High

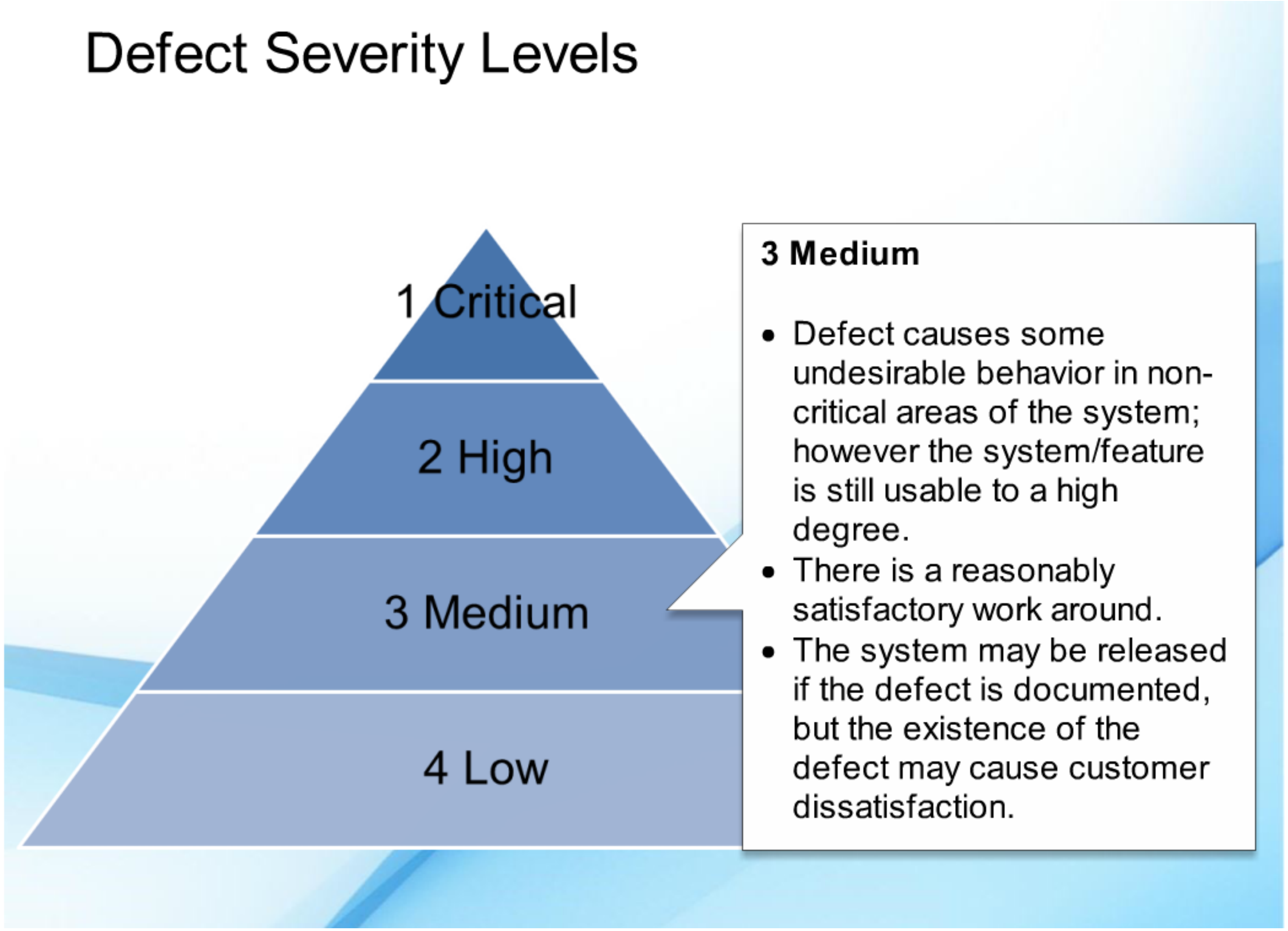
3 Medium

4 Low

2 High

- Defect results in severely impaired functionality.
- While there may be a workaround for using the system, its use in production is unsatisfactory.
- A high severity defect will prevent the product from being released if not fixed.

Defect Management Process

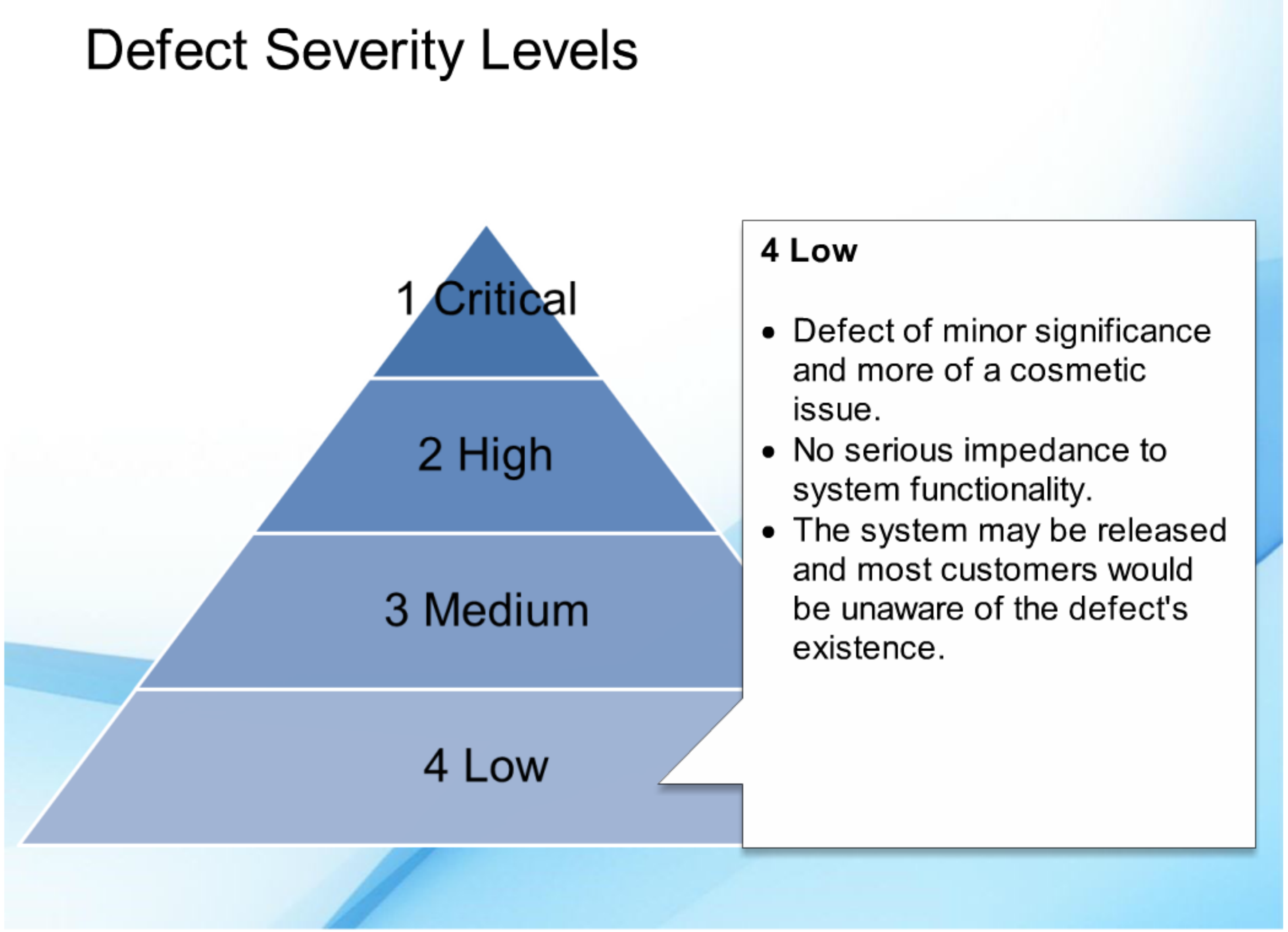


Defect Severity Levels

3 Medium

- Defect causes some undesirable behavior in non-critical areas of the system; however the system/feature is still usable to a high degree.
- There is a reasonably satisfactory work around.
- The system may be released if the defect is documented, but the existence of the defect may cause customer dissatisfaction.

Defect Management Process



Defect Severity Levels

4 Low

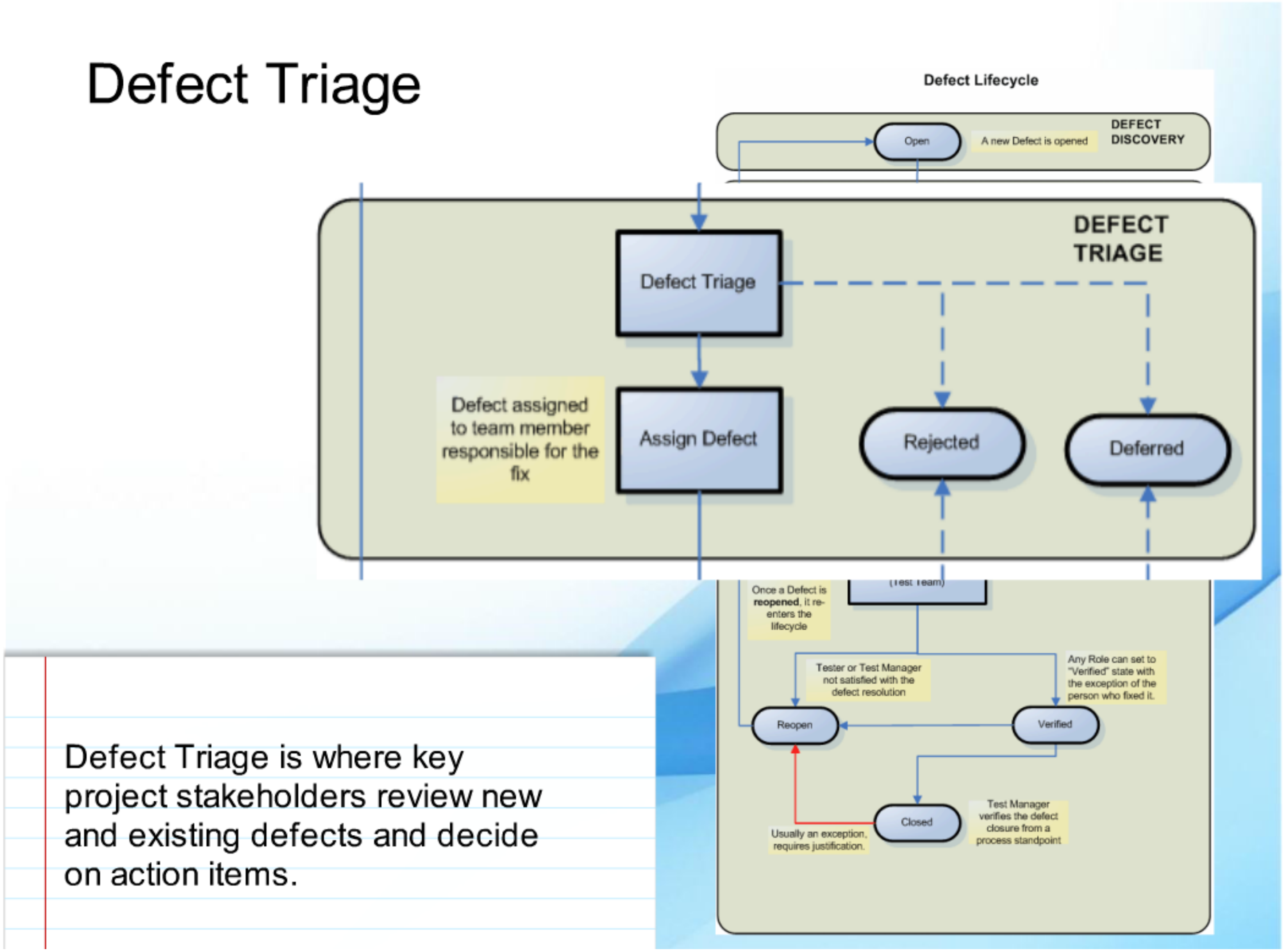
- Defect of minor significance and more of a cosmetic issue.
- No serious impedance to system functionality.
- The system may be released and most customers would be unaware of the defect's existence.

Defect Management Process

✓ **Defect Triage**

Defect Triage is a process where key project stakeholders review new and existing defects and decide on action items. These defect triage meetings can occur everyday if project size and frequency of defects warrant it, but at a minimum they must be triaged during the iteration to bring efficiency to defect resolution and allow for planning of fixes in a successive iteration.

Defect Management Process



Defect Triage

Defect Lifecycle
Open
A new Defect is opened
DEFECT DISCOVERY
DEFECT TRIAGE
Defect Triage
Defect assigned to team member responsible for the fix
Assign Defect
Rejected
Deferred
(Test Team)
Once a Defect is **reopened**, it re-enters the lifecycle
Tester or Test Manager not satisfied with the defect resolution
Any Role can set to "Verified" state with the exception of the person who fixed it.
Reopen
Verified
Closed
Usually an exception, requires justification.
Test Manager verifies the defect closure from a process standpoint

Defect Triage is where key project stakeholders review new and existing defects and decide on action items.

Defect Management Process

At minimum, Michael the Project Manager, Angela the Test Manager and Stefan, the Development Lead attend the Defect triage meeting, but anyone on the project team can attend.

During the defect triage meeting, new defects are evaluated, and any clarifying questions are answered by the Test Manager. The triage team performs an impact analyses and assigns a priority for each defect based on the urgency at which the defects need to be fixed. The severity that was previously defined by the tester will also be discussed and may be changed as long as the rationale for the change is documented in the defect's comments section.

✓ **Defect Triage Outcomes**

There are three outcomes for a defect in the Triage stage: Assigned, Rejected and Deferred.

Assigned: If a defect is determined to be valid, it will be **assigned** to the team member responsible for fixing it.

Rejected: A defect can be **rejected** by the Test Manager if the defect determined to be invalid or a duplicate. Rationale for rejected defects must be documented in Application lifecycle management.

Deferred: A defect can be **deferred** to a future release, but it is not considered good practice and should be treated as an exception rather than the norm. The rationale along with a clear articulation of risk and impact must be documented in Application lifecycle management.

Defect Management Process

✓ **Defect Deferral**

Defect Deferral, while one of the possible outcomes during the defect triage process, is not a decision that should be made lightly and definetly not used as a way to meet a delivery deadline.

There are a specific criteria to determine if a defect is eligible to be deferred. The testing work instruction(IT225) provides details on the conditions which must be met to defer a defect. The customer representative must also understand the specifics of the deferral and provide their approval.

Each potential deferred defect must go through an impact analysis where the risk and proposed actions are discussed. Only defects with low risk, and minimal to no impact to system functionality or user experience can be considered for deferral. If a defect is associated with a regulatory requirement, or has a severity status of Critical or High, then it cannot be deferred.

Deferred defects that have the potential to impact user experience require business approval. The approval and rationale for the deferral, including any work-arounds must be documented in Application lifecycle management.

If a defect is deferred, then it must be managed in a controlled manner to ensure it is addressed in a future release. Every deferred defect must have a corresponding backlog item with a unique ID that can be used for tracking.

The Test Manger, will document in the comments field any pertinent information from the defect triage discussions that can be useful for tracking defect history.

Defect Management Process

In addition, any defect with a severity listed as Critical, or tied to a regulatory requirement will require a Root Cause Analysis discussion and the results documented in Application lifecycle management.

Angela assigns the defect to Seema, a developer.

When Seema starts work on the defect, she first changes the status to In Progress.

Seema performs the coding to fix the defect. She updates the status in the defect when she is complete.

✓ **Defect Resolution**

Defect Resolution is meant to bring defects to closure, either through a closed status, or a rejected or deferred status. All the defects must be in one of those states prior to production deployment.

Defect Management Process

When getting to a defect resolution is the primary outcome of this process, tracking a defect through its lifecycle, from discovery, through triage and to resolution is an equally important part of the defect resolution process.

VJ, the tester reviews the defect Seema has fixed. If he is not satisfied with the fix, then he changes the status to **Reopen**, updates the comments, and the developer would have to work on it again.

If he can replicate the fix, he then changes the defect status to **Verified**.

As the Test Manager, Angela reviews defects with the **Verified** status.

If she finds any issues with the defect, she can change the status to Reopen, update the comments, and it goes back through the defect management process. However, if she agrees with the tester's findings, she changes the status to **Closed**.

Now that you have seen how the defect management process works, you will now go through the steps in Application lifecycle management to take a defect from discovery through resolution.

✓ **Defect Discovery**

Open a Defect: VJ, the tester is executing a test set when he comes upon a defect. In this module you will open a defect.

Defect Management Process

Click the **Fail Selected** menu.

Enter the actual result.

Click the **New Defect** button.

Enter the Defect Summary.

Update the **Found During** field. Click **Pre-Validation.**

Click the **Severity** dropdown button. Click **2-High** severity. Click on **Ok**.

Click the **End Run** button.

If the defect discovered does not prevent you from completing the test, you can move on to the next step. If the tester is not able to continue with the test steps, tester can decide to end the test.

✓ **Defect Triage**

Angela, the Test Manager has just completed the triage meeting and now needs to assign the valid defects to the production team member responsible for addressing the defect. In this module you will assign the defect.

Defect Management Process

Click **Defects**.

Click the **Assigned to** dropdown button. Select a resource. Click **Ok.**

Click the **Priority** dropdown. Click the **2-High** priority. Click **Ok**.

✓ **Fix a defect**

Seema, the developer is assigned the defect and has the responsibility to fix it. In this module you will see how to use Application Lifecycle management to track the progress of fixing a defect.

Click the defect.

Click the **Status** dropdown menu. Click **In Progress**.

Click the **Components** dropdown. Click **User Interface**. Click **Ok**.

Seema has fixed the defect and now needs to update the defect record in Application lifecycle management.

Click the defect.

Defect Management Process

Click the **Status** drop down. Click **Fixed**.

Click the **Defect Root Cause** drop down. Click **Code_Source**.

Click the **Code_Source** dropdown. Click **WebPart**. Click **Ok**.

✓ **Defect Resolution**

Verify a Defect – The Developer corrected the defect and changed the status to Fixed. Now VJ can go back and verify that the defect has been fixed.

After he confirms that the error is gone, he can update the defect in Application lifecycle management.

He will begin by filtering the list of defects to only show items that have the status as Fixed.

Click the Set **Filter/Sort** button.

Click the **Status** dropdown menu. Click **Fixed**. Click **Ok**.

Click the Defect.

Defect Management Process

Click the **Status** drop down. Click **Verified**. Click **Ok**.

✓ **Close a Defect**

Defects can only be closed by the Test Manager. Angela will review all defects that have the Verified status and are accurately completed before she closes the defect.

She will begin by filtering the list of defects to only show items that have the status as Verified.

Click the **Set Filter/Sort** button. Click the **Set Filter/Sort** item.

Click the **Status** dropdown. Click **Verified**. Click **Ok**.

Click the defect. Click the **Status** dropdown menu. Click **Closed**. Click **Ok**.

Defect Management Process

✓ **Summary**

You should now be able to:

Define a defect.

Identify the three phases of the defect management process.

Compare the four defect severity levels.

Use Application lifecycle management to a take a defect through the defect management process.