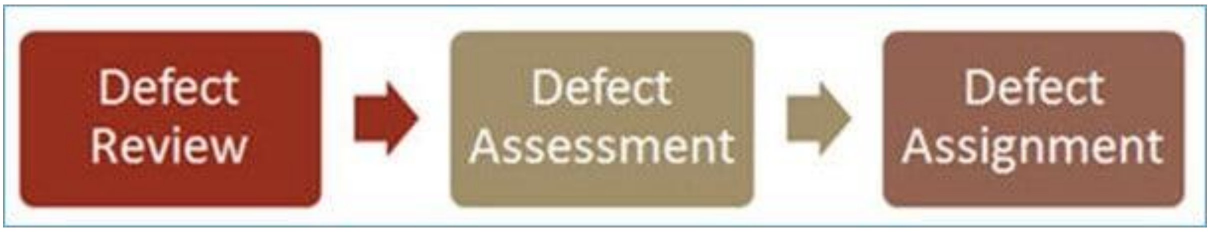
Bug/Defect Triage in Software Testing

What is ‘Defect Triage’?

Defect triage is a process where each bug is prioritized based on its severity, frequency, risk, etc. Triage term is used in the Software testing / QA to define the severity and priority of new defects.

Why do we need to have ‘Defect Triage’?

The goal of Bug Triage is to evaluate, prioritize and assign the resolution of defects. The team needs to validate severities of the defect, make changes as per need, finalize resolution of the defects, and assign resources. Mainly used in agile project management.



How often ‘Defect Triage’ needs to be conducted in a release?

The frequency of Defect triage meeting is not fixed. It depends on project situation.

Here, are some important factors that decide the frequency of Defect Triage Meetings:

These Important factors are:

- As per the project schedule
- Number of defects in the system
- Impact on schedules of team members’ availability
- Overall project health

Usually, Defect Triage Meetings are held two or three times in a week.

Who are the mandatory and other participants of ‘Defect Triage’?

Mandatory Participants

Below project members always take part in Defect Triage Meetings.

- Project Manager
- Test Team Leader
- Technical Lead
- Development Team Leader

Optional Participants

- Developers
- Testers
- Business Analyst

Roles and Responsibilities of participants during ‘Defect Triage.’

Test Team Leader

- Scheduled bug triage meeting and send meeting notification for attendees.
- Create a defect report and send it to all attendees before the meeting.
- Assign priority and severity of the defects.
- Give a presentation so that other members understand Root Cause of defect.
- Every meeting note is captured and sent to meeting attendees.

Development Lead

- Helps in the prioritization of the defects.
- Discuss defect difficulty and explain the risk involved because of that defect.
- Allocate work to fix defects to relevant developers.
- Update the defect resolution and include development notes in case any information is missing or any additional information needed by developers.

Project Manager

- Help in the prioritization of the defects.
- Discuss the next iteration release date for QA.
- Need to make sure that related user representatives are also invited to the bug triage meeting.

What happens during ‘Defect Triage’ Meeting?

- Test Team leader sends out a bug report with the new defects. During the defect triage meeting, each defect is analyzed to see whether right priority and severity are assigned to it.
- Priorities are rearranged if needed.
- Defects are analyzed and evaluated by the degree of their severity.
- This include discussion regarding the complexity of the defect, risks, rejection, reassignment of errors is done.
- Updates are captured in bug tracking system.
- The QA engineer will make the changes to each defect and discuss them with each attendee.
- The “Comments” field is updated correctly by noting essential points of the meeting.

What is the outcome of the ‘Defect Triage’?

At the end of every meeting, Defect Triage Metrics will be prepared and given to all the attendees. This report acts as the meeting minutes which will prove helpful for future meetings.

Conclusion:

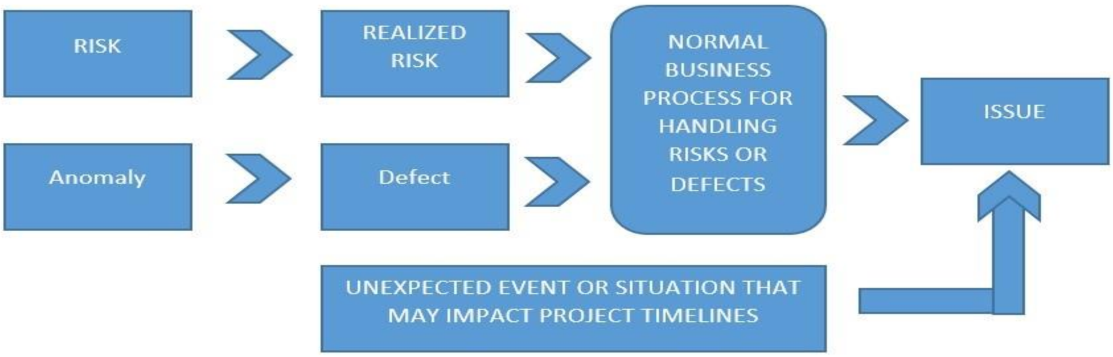
- Defect triage is a process where each bug is prioritized based on its severity, frequency, risk, etc.
- The goal of Bug Triage is to evaluate, prioritize and assign the resolution of defects.
- The frequency of defect triage meeting is decided according to the project schedule, number of defects in the system, overall project health, etc.
- Project Manager, Test Team Leader, Technical Lead, Development Team Leader are taken part in this meeting.
- Defects are analyzed and evaluated by the degree of their severity.

Risk, Anomaly, Defect, or Issue? *Does it matter?*

One of the most basic functions of Testing and Quality Assurance activities is documenting unexpected outcomes and ensuring that anything done to resolve a problem is satisfactory. Understanding the difference between Risks, Anomalies, Defects and Issues is important to establishing the protocols used by your project in resolving them.

I like using the metaphor of the Roadside Deer to illustrate these basic concepts. Put simply, **a deer on the side of the road is a Risk** for drivers. It may jump out in front of your car, it may not. But just *knowing* that there are deer in the woods enables you to start planing. What can we do to make sure the deer doesn't jump out in front of the car? This is called **Risk Mitigation**. Some people use direct methods, like putting a whistle on the bumper, theoretically emitting a sound that repels deer. I don't know if these actually work, but that's another article. Another direct method of Risk Mitigation might be the use of powerful headlights, enabling you to see the deer farther in advance giving you more time to react. A business parallel to this might be routine reports, enforced Project Stop Gates, or Risk Review sessions. Some people use indirect methods, i.e. **Risk Avoidance**. They know that dawn and dusk are times where deer are *more likely* to be on the side of the road, so they avoid those times. Or they drive more slowly, giving time to act in the event a deer is spotted. Drive slow enough, and there is no Risk.

What if the deer actually does jump out in front of your car. Is that an Issue? Only if your Risk Management Protocols are unable to counter it without making you change the overall plan. No, the Deer in the Road is called a **Realized Risk**, sometimes referred to as a Threat. You knew it might happen, and it did. The amount of planning you did in advance usually determines the overall impact to your Project. The game of **Risk Management** continues with **Contingency Planning**. This is what you will do if and when the Risk is encountered. Step on the brake. Swerve. Plow right on through. The same goes for your Project. Each option has pros and cons, and discussing them in advance will have the effect of reducing the overall impact to Project Timelines. By having plans in place, it will be possible for Project Managers to immediately roll out a new set of Tasks designed to deal with it. With adequate planning, many Realized Risks can be eliminated before they become worse.


RISK
REALIZED RISK
NORMAL BUSINESS PROCESS FOR HANDLING RISKS OR DEFECTS
ISSUE
Anomaly
Defect
UNEXPECTED EVENT OR SITUATION THAT MAY IMPACT PROJECT TIMELINES

If the Contingency Plan for any given Risk was unable to overcome the issue when encountered and additional planning is required, the Risk automatically becomes an **Issue**. More on that later.

Which brings us to an **Anomaly**. An Anomaly is anything encountered in testing (or routine software use) that is an Unexpected Result. Anomalies require immediate documentation, so that appropriate users can make a determination if there is anything that needs to be fixed. This is called **Vetting**. User error? Maybe it's important enough to improve training. Software bug? Can it be repeated? But I'm getting ahead of myself. Have you ever driven down the road and been *certain* you saw a deer on the side of the road, only to discover it was actually a mailbox? That's the Anomaly. But it has nothing to do with the deer - it has to do with how the driver (you) reacted in an environment where there might be deer. Maybe the solution is just better training or more experience.

A **Defect** is something that needs to be fixed. Software Developers are usually OK with this term, though they might prefer calling it a 'Bug'. System Configuration professionals will balk at the term 'Defect'. They like to think of those problems as 'errors', not 'defects'. Just a typo, right? Data Migration and Interface resources might similarly cringe at the term 'Defect'. Let's get past the word, folks. All it means is something that needs to change. The fix can be manual, it might require programming, it might require data cleanup, it might require modification of mapping, business rules, or a User Interface - or any number of other solutions.

Is a Realized Risk or Anomaly the same thing as a Defect? No. This is not just splitting hairs. A Realized Risk and a Defect are fundamentally different in one way: The Defect is something that needs to be *fixed*. The Realized Risk is something that needs to be *overcome*. A Risk Management Plan and a Test Plan will generally describe the normal process for handling the two, and the people and methods employed are usually different.

Once an Anomaly has been vetted, it is usually labeled a Defect. But a Realized Risk should never be thought of as a Defect. It's important to have a **Defect Management Protocol** established to predetermine the communications, documentation standards, responsibilities, response times and roles so that Defects are quickly resolved by the right people, retested, and Closed. We'll revisit the deer metaphor after discussing Issues.

The deer on the side of the road is a Risk. The deer in the road is a Realized Risk. If you can't avoid it in time - that's an Issue.

An **Issue** is something that occurs that cannot be dealt with within the guidelines of the Risk Contingency Plan or the Defect Management Protocol. Example: A configuration error needs to be resolved within 48 hours of notifying the appropriate party that a change is needed. If one is found and corrected within normal parameters, then it's not an Issue. But sometimes reality skates in with a hip-check (hockey metaphor) and this simply isn't possible. Maybe business decisions need to be revisited. Maybe workflows need to be redesigned. Maybe there are contract delays, or a key resource has left the company. These become Issues, and there are only two kinds. Those that MIGHT affect project timelines, and those that WILL affect project timelines. Issue Management is more difficult, expensive and time-consuming than Defect Management because it involves after-the-fact planning using high-level resources. Similarly, Risk Management is much less expensive than Issue Management. A key planning error of many projects is ignoring Risk Management, or not allowing enough time to adjust for routine problems.

Or, an Issue may come out of the blue. The hotel you always use for training burns down. Not being able to secure a training room big enough to accommodate your staff may be an Issue for the project and may affect timelines, but it has very little to do with any 'Defect' that may be encountered through testing or system use.

The deer on the side of the road is a Risk. The deer in the road is a Realized Risk. If you can't avoid it in time - that's an Issue.

So where is the Defect in our little deer-in-the-headlights scenario? It has nothing to do with the deer. Remember that Defects are something you encounter while testing. The deer is something that may prevent you from testing, but it's not something you need to *fix*. You just need to *deal* with it.

So what does this mean for Project Management and Test Planning?

First, establish clear guidelines in your Master Test Plan for how Anomalies get vetted, and by whom. Establish documentation standards. Formalize the Defect Management Strategy. Establish Priority protocols for ranking Defects, and guidelines on how long is an acceptable period for resolving different types of Defects. Establish a clear guideline on when a Defect must be re-documented as an Issue and who is involved in that process.

Second, work with the Project Manager and cross-team facilitators to set up routine scheduled meetings for Defect Management and Issues Management. Also - set up meetings early in your project to talk seriously about Risk identification, strategic planning, and overall Risk Management. Some companies will look at this last as 'borrowing trouble', but Risk Management guidelines can be reused from project to project, and they do pay dividends.

Third - stay true to your guidelines. Too often people approach Issues and Defects as though everything is 'Critical'. It's not. And overuse of the word 'Critical' guarantees that it will eventually be ignored, rendering it useless for making determinations of Go/No-Go, or even prioritizing work effort during the project.

The difference between Risks, Anomalies, Defects and Issues is important. Each requires different planning, and involves different resources at different times during the Software Development Life Cycle. It is possible and even advisable to combine Anomalies and Defects during the documentation phase using appropriately controlled timelines and metadata, but managing the communications is easier if you separate the Defects, Risks and Issues, because they really are very different.