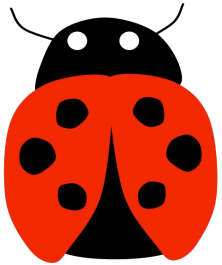
We are

Globant



Bugs Management

Remote Ice Breaker

If you had to delete all but three apps from your smartphone, which ones would you keep?

Today's Bugs Agenda

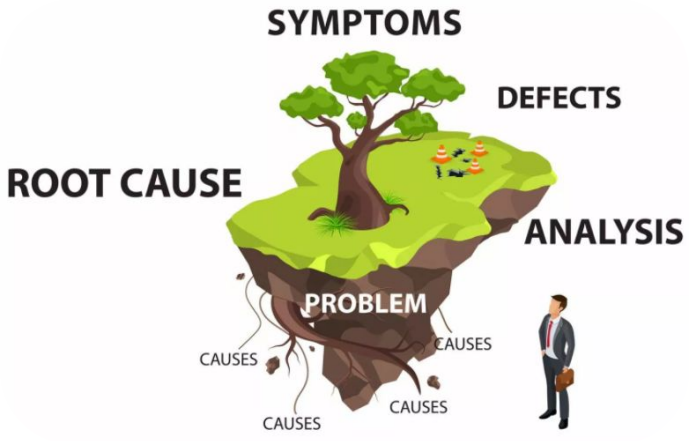
- Bug's Definition.
- Root cause analysis (RCA).
- Components.
- Bugs Life cycle.
- Example of a bug.
- Good practices.
- Bugs tracker tools.
- Questions

Bug's Definition

- You might observe that the actual results is different from expected results.
- When the actual result is different from the expected result then it is called as incidents, bugs, defects, problems or issues.
- Incident vs bug/defect:
  - **Incident:** Any situation where the system shows unexpected behavior.
  - **Bug/Defect:** Incidents are bugs/defects only when the root cause is a problem in the item we are testing.
  - **Failure:** After release, if an end user finds an issue then that particular issue is called as failure.

Root cause analysis (RCA)

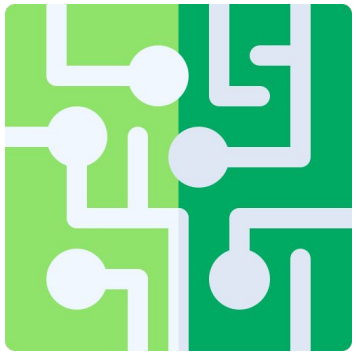
- Mechanism of analyzing the Defects, to identify its cause.
- We brainstorm, read and dig the defect to identify whether the defect was due to:
  - “testing miss”
  - “development miss”
  - “requirement or designs miss”
- It helps to prevent defects in the later releases or phases.



Root cause analysis (RCA)

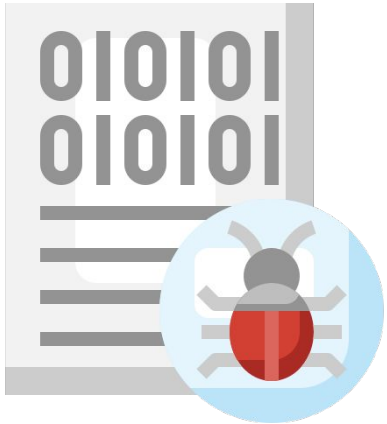
- There are many factors which provoke the Defects to occur:
  - Unclear / Missing / Incorrect Requirements
  - Incorrect Design
  - Incorrect Coding
  - Insufficient Testing
  - Environment Issues (Hardware, Software or Configurations)
- Example: Suppose that the calculation of account interests is incorrect, due to a single line of incorrect code, they give rise to customer complaints, because the requirement was ambiguous.

What are the components of a bug?



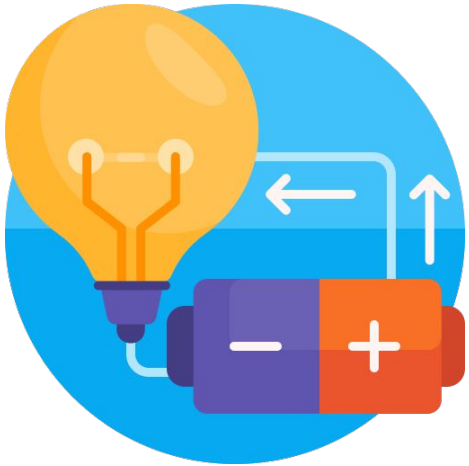
Bug Id

- A code that allows you to unequivocally identify the bug.
- It is usually generated automatically by the bug tracking tool.
- It can be used to link the defect with the failed test case.



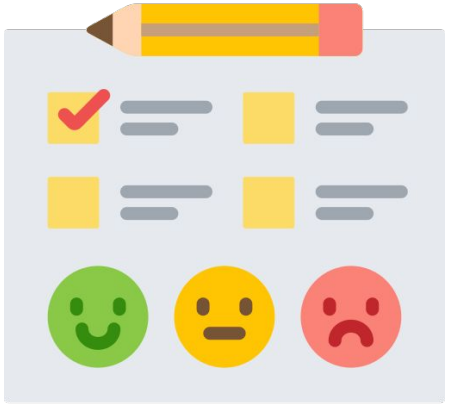
Summary

- Reduced description of the identified problem
- You should try to describe the problem without going into details



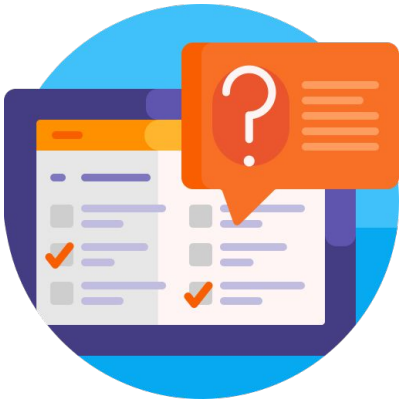
Description

- Detailed description of the problem identified.
- It includes:
  - Description of the problem
  - Precondition
  - Steps to follow to reproduce
  - Input data used
  - What should be the expected behavior
  - Mention of the US or requirement that describes the correct or expected behavior.



Severity

- Indicates the impact that a defect has on the development or operation of a component or system.
- In general, the level of severity is linked to the tool used to track the bugs. Example: **Low, Medium, High, Urgent, Crash**
- Each project / team can define what each level of severity means according to the system to be tested or the business rules.



Priority

- Level of importance (commercial) or impact on the business that the bug has.
- It may be related to the severity of the bug but it is not always the case.
- Sometimes the priority is not defined by the QC Analyst but business analysts, project managers or Product Owner participate.

Severity VS Priority

Severity/Priority	Low Priority	High Priority
Low Severity	Bug with no big impact that can wait to be resolved	Bug with no big impact that needs to be fixed urgently
High Severity	Serious bug that does not require fast resolution (Unlikely)	Serious bug that needs to be resolved urgently Probably blocks other tasks

Severity Vs. Priority

- **High Priority and High Severity:** The system crashes after you made the payment
- **High Priority and Low Severity:** Logo of the company in the front-page is wrong
- **High Severity and Low Priority:** A beta version of a new feature is released with not many active users using, and the user can not select a product.
- **Low Severity and Low Priority:** If the privacy policy of the website has a spelling mistake.

Additional information

- Here you can add more details that help dev team or any other person to reproduce and also fixed the bug.
- What Can We put there? Examples:
  - It is happening in other modules.
  - It reproducible in a particular operating system or browser.
  - The bug is related to another bug that already exists (link them).

Attachment

- Any **file** that you attach can help to the dev team to understand and then fixed the bug.
- In order to get a better understanding, it is possible to attach things like **screenshots**, **videos**, **gif**, etc.
- **Logs** will be so useful in order to get extra information.

EXAMPLE

Bug ID: 26438	Category: product	Severity: Medium	Priority: High

Summary: When executing a search the results are not ordered by price

Description: When executing a search the results are not ordered by price and should appear ordered from lowest to highest by default.

Steps:

1. Select restaurant option
2. Select a restaurant
3. Verify the order of the offered products.

Current Result:

The results are not ordered by price

Expected Result:

The results, according to the US, should be ordered by price, from lowest to highest

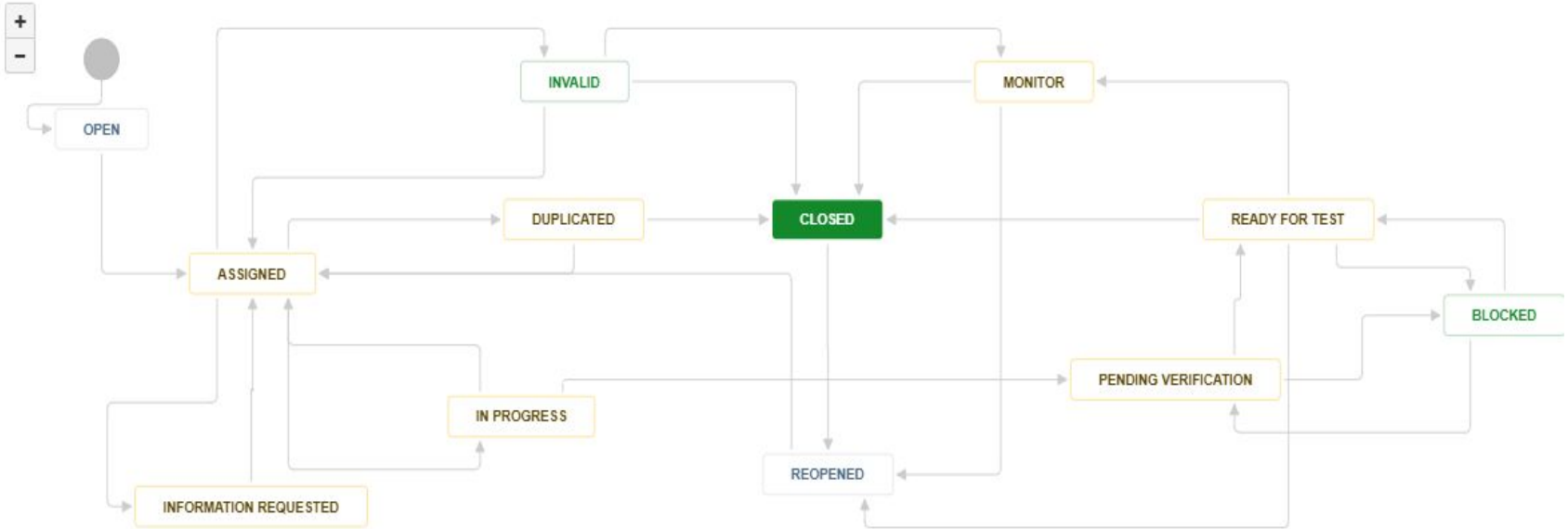
Additional information: This is happening in all restaurants. Only discounts in Chrome, in Firefox mobile the search results are ordered as expected.

Attachment: screenshot1.jpg

What statuses can a bug have?

- Open: When a tester raise a new bug.
- Assigned: When a dev person takes a bug that was open or reopen, or the technical leader assigned to him
- IR (Information Request): When the developer has an issue trying to reproduce the bug or needs more info.
- In progress: When the developer is going to start fixing the bug.
- Rejected: When the developer finds that this scenario is invalid, or the complexity and time to invest there against the severity does not have sense, or it is not a defect is the expected behavior from the customer side.
- Resolved: When the developer has a fix for that bug.
- Closed: When the tester verifies that the bug was correctly fixed.
- Reopened: When the tester verifies the result and the bug is still happening.
- Monitor: When a bug is being investigated by all groups, or simply pending of certain validations.

Bugs – Life cycle (Sample)



Bugs Example

Let's check this bug

The system allows me to register as a user, by entering an email address that has been previously registered and should not.

Bugs – Example of creating a bug

What is missing?

Bug ID: 26348	Category: User Registration	Severity: High	Priority: Medium

Summary: Multiple users can be registered with the same email address

Description: The system allows me to register as a user by entering an email address that has been previously registered and should not. Instead of the current behavior the system should show me the error message *"A user with that email already exists"* and return the User and Email fields to their initial state.

Additional information: created users

User: pperez
Email: pperez@gmail.com
Password: f12bc69d-a61e-4f68-f

User:pperez
Email: pperez@gmail.com
Password: f12dl92d-894f-4g73-z

Attachment: none

To report bugs... Here a guide!

id	1
Title	Short description of the behavior
Environment	QA; PROD; UAT
Priority	hight
Severity	blocker
Description	Brief description about the behavior of the system when the user does a specific action
Precondition	This means that is important before executing the test.
Steps:	1. Go to Url <>
	2. Click on the option N
	3. Select an other option
Current result	This is the actual behavior of the system when the user does something specific
Expected result	How the system should behave when the user does that specific action
Evidences	Picture.jpg



Bugs – Best Practices

A bug should have enough information as possible

A bug should include:

- Description of the problem with steps about how can reproduce the issue
- Current result vs expected result

Bear in mind:

- Reproduce the defect at least twice
- Figure out the impact correctly to set the severity and priority*
- The summary should be only one clear sentence

Complete definition of the categorizing fields:

If you work with a Structured Bug tracking system you will have fields that categorize your issues (Module, Infrastructure, Browser, etc) bear in mind the mandatory fields.

Set the Correct Severity & Priority:

A big Tester sin is to over prioritize your bugs

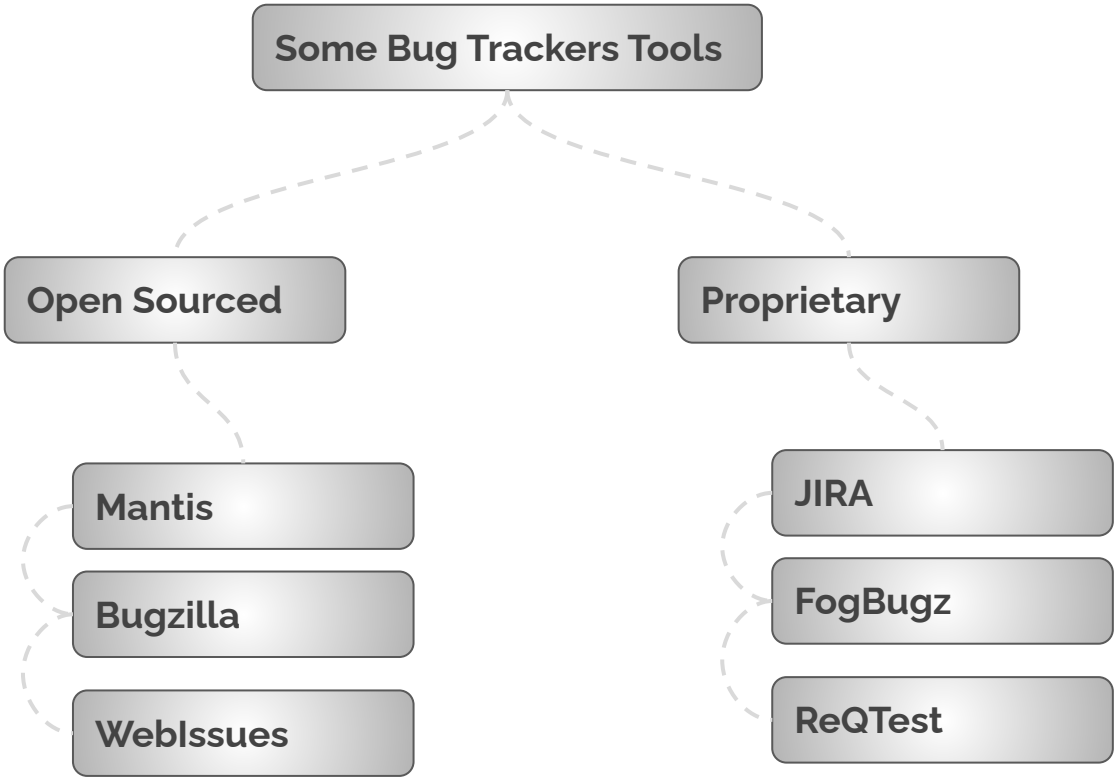
Follow-up:

You should continue following up on your bugs, and providing comments when you need it

Evidence:

Always use an evidence (screenshot, video, logs). Specially when the bugs is related to UX.

Trackers Tools



QUESTIONS

Thank You!