Test Execution and Reporting

SENG 5811 – Spring 2023

Week #9

Software Defect Reports/Test Reports

- Software Defect Reports
- Defect Tracking Systems
- Test Reports

Software Defect Reports

Software Defect Reports

- **Why**: The purpose of testing is not only to find errors, but to see that they get fixed. The Defect Report is the principal means of communicating with Development in a repeatable, organized way.
- **Take-away**: Understand the life cycle of a Defect Report, and the components of a Defect Report.

“To ensure reliable and fast elimination of failures detected by the various test levels, a well-functioning procedure for communicating and managing those incident reports is needed.”

Another author’s caution:

“ *Independent Testing* does not mean that we work in isolation from the developers.”

Handling a software defect

- **Analyze the defect**
- Report the defect
- Track the defect
- Retest the defect
- Close the defect

Synonyms: SUR, UCF, PTR, SCR, Incident Report

Analyzing a software defect

- Is it the test, the tester, or the product*?
- Reproducible vs Repeatable (or Repeating)
- Can it be isolated?
- Look for alternate paths to the problem
- Is it worth reporting (formally)?

* or an environmental or hardware failure

Handling a software defect

- Analyze the defect
- **Report the defect**
- Track the defect
- Retest the defect
- Close the defect

Reporting the software defect

Once you know it's a reportable issue:

- Look for open duplicates
- Talk with the developer
- Enter it into the system
- Make sure it will get noticed!

Characteristics of an effective software defect report

- Tells how to reproduce the problem…
- … in a minimum number of steps
- The report is complete and understandable
- Non-antagonistic
- Written (not just a verbal “handshake”)
- Simple – one issue per defect report
- Numbered/ID’d

Components of a software defect report

- Identifying information
- Description of the problem
- Various status indicators
- Comments
- Miscellaneous information
- Supporting information

Identifying information [for Lab #3!]

- **Identifying information**
- Description of the problem
- Various status indicators
- Comments
- Miscellaneous information
- Supporting information

Unique number or ID

Submitter

Submit date

Program or product this is against

Version or revision of the product

Description of the problem [for MP #2]

- Identifying information
- **Description of the problem**
- Various status indicators
- Comments
- Miscellaneous information
- Supporting information

- Title
  - Must convey in a limited number of words enough information that the problem can be understood
  - Any abbreviations must be generally acknowledged or explained in the Description
  - Example: "ACM: No error messages when 'Print on Error' is on"
- Description (following slides)

- Identifying information
- **Description of the problem**
- Various status indicators
- Comments
- Miscellaneous information
- Supporting information

Description: Contents (I)

- The problem itself
  - What did happen (actual results)
  - What should have happened (expected results)
  - Ex.: "I left the final delimiter record out and reports got produced. I expected an error message and no reports."
- The test case used
- Any other helpful information whatsoever
  - In particular, *name* any attachments
- Keep it to 3 or 4 sentences if you can

- Identifying information
- **Description of the problem**
- Various status indicators
- Comments
- Miscellaneous information
- Supporting information

Description: Contents (II)

Avoid	Avoid vague or confusing terms (“frequently”, “sometimes”)
Avoid	Avoid uncommon abbreviations
Use	Use any standard terminology
Watch	Pay attention to spelling and grammar

- Identifying information
- Description of the problem
- **Various status indicators**
- Comments
- Miscellaneous information
- Supporting information

Status indicators

In my experience, we have used:

- Overall defect report status
- Severity and priority
- Current resolution status

- Identifying information
- Description of the problem
- **Various status indicators**
- Comments
- Miscellaneous information
- Supporting information

Report Status

- The entire defect report is one of
  - Open
  - Closed

- Identifying information
- Description of the problem
- **Various status indicators**
- Comments
- Miscellaneous information
- Supporting information

Report Status

The entire defect report is either …

- Open
- Closed

- Identifying information
- Description of the problem
- **Various status indicators**
- Comments
- Miscellaneous information
- Supporting information

Severity vs Priority

Severity: How "bad" is it? (Next slide)

- Generally related to the defect's effect on testing

Priority: How urgent is it?

- When does this have to be fixed?

- Identifying information
- Description of the problem
- **Various status indicators**
- Comments
- Miscellaneous information
- Supporting information

Severity

High -- critical defect; test cannot continue

Medium -- critical defect; test can continue with a workaround

Low -- it's a problem but we can essentially ignore it

Really low -- documentation, help files, spelling mistakes

- Identifying information
- Description of the problem
- **Various status indicators**
- Comments
- Miscellaneous information
- Supporting information

Examples of Resolution status (interim)

- None [this is what the board searches on]
- In Process/Assigned
- Fixed ("pre-build" for an app defect)
- Ready for Test ("post-build"; not used for a defect in the Test Suite itself)
- Retest

  For a typical defect, these 4 are all we need, plus 'Closed' [and the overall 'Closed']. Note too – these are searchable.

- Identifying information
- Description of the problem
- **Various status indicators**
- Comments
- Miscellaneous information
- Supporting information

Examples of Resolution status [final]

Not repeatable

Not a problem

Do Not Fix

Duplicate (probably shouldn't have been entered at all)

Deferred

Comments/Notes

- Identifying information
- Description of the problem
- Various status indicators
- **Comments**
- Miscellaneous information
- Supporting information

ANALYSIS

RESOLU-
TION

TESTER

**MODIFI-
CATION**

**VERIFI-
CATION**

WHATEVER-
WORKS-
FOR-YOU
NOTES

Miscellaneous fields (I)

- Identifying information
- Description of the problem
- Various status indicators
- Comments
- **Miscellaneous information**
- Supporting information

- Steps to Reproduce
  - Include setup information
  - Anyone with the appropriate setup should be able to reproduce the problem
  - Could be as simple as a Test Case ID, or as complex as a lengthy procedure
  - Pay attention to problems that are merely repeatable and not reproducible

Miscellaneous fields (II)

- Identifying information
- Description of the problem
- Various status indicators
- Comments
- **Miscellaneous information**
- Supporting information

Environment	Discovered by
Target Release	Defect type/Root cause
Closed Release	Software component
Closed Date	Fix hours
Owner	Test hours
Tester	Workarounds

- Identifying information
- Description of the problem
- Various status indicators
- Comments
- Miscellaneous information
- **Supporting information**

Supporting information

- Error printouts
- Screen shots
- The test case itself
- A flash drive with data or files
- Trace files, error logs, etc.

The Rest of Handling a Defect

Analyze the defect

Report the defect

Track the defect

Retest the defect

Close the defect

Tracking a Defect

Must have a process in place

Generally done by some sort of Review Board

The lead tester should be on that board

The purpose is to make sure the defect progresses from being submitted to getting closed.

Retesting a Defect

Retest can have 3 outcomes*

- The problem has indeed been fixed
- The problem remains
- A different problem shows up

* not necessarily mutually exclusive!

Closing a Defect

Add	Tester Notes or Verification Notes
Inform	Anyone else with a vested interest in the problem's resolution so that they have a chance to test it too
Close	The report, or recommend closure

Summary: Life cycle of a software defect report

- Entered by a tester (usually)
- Review Board evaluates and assigns information
- The assigned developer fixes the problem
- A build containing the fix becomes available
- The tester retests the problem
- The tester or board closes the defect

Defect Tracking Systems

Defect Tracking Systems

- Why:

  As noted, defect tracking must be written, not verbal.

- Take-away:

  Understand the characteristics of a good defect tracking system.

The need for a Defect Tracking System

- Facilitates clear communication about defects
- Allows developers to fix bugs based on priority
- Allows for analysis and reporting options
- Helps to manage the bugs
- Keeps a history

Characteristics of a good bug tracking system

Automatic numbering

Configurable in several ways: field contents and what you see, when

Comes with a report generator

Allows for user restrictions

Allows for suggestions

Fields can be marked as ‘mandatory’

General comments

- Remember: The main purpose of a bug tracking system is to get bugs fixed
- We also use the information to generate reports that help track progress
- Make sure no error goes unfixed because someone forgot about it

Caveats

- If it's not your bug, but you have to test it, make sure you can repeat the problem first.
- Tracking test cases to modules or components can be helpful.
- Never measure anyone with bug-count statistics.
- Anyone can enter a problem report.
- Testers **can** hold up a product release.

A Comparison of Defect Tracking Systems

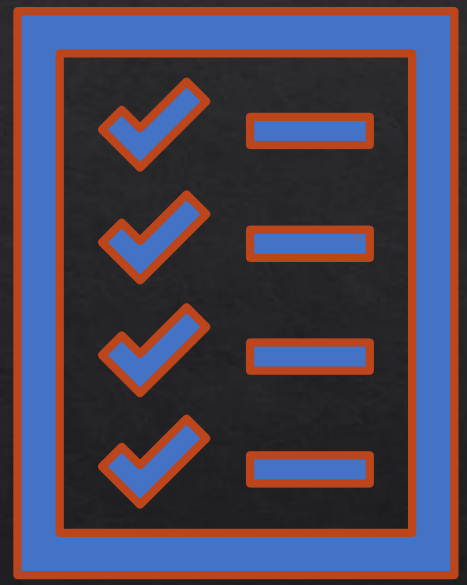
https://www.capterra.com/bug-tracking-software/

Dated 2019. Includes Bugzilla, IBM Rational, FogBugz, Jira, and many more I've never heard of.

I have used:

- PVCS Tracker;
- JIRA (Atlassian)

The Test (Status) Report



Take-Aways

- Test Report
  This is where we tell the world how testing went or is going.
  - Test Status/Progress Report
  - Test Summary Report

  - Understand what goes into a Test Report.

Definitions

- Test Status Report
  - Tells how the current test cycle is going
  - Our text recommends a Test Status Report after each testing cycle
- Test Summary Report
  - Tells how the entire test effort went for a project or feature
  - Written after testing is done

The following slides address both reports simultaneously.

Test Reports

- Your chance to evaluate how testing went, or for a status report, how it's going
- List what was tested
- List what was not tested, and why
- List still-open defects (some ask for all defects)
- Show the actual schedule (from the Test Plan)
- Tell developers what works and what doesn't work
- Lets management know the status of the product, from a testing standpoint

Objectives of Test Reports

Allow management, marketing, etc., to make a determination as to whether or not the system is ready for production/release, or will be ready on time

Assist in process improvement – both the Software Development Process and the Testing Process

Your Test Report

- Cover sheet with Team Name and student names
- A brief description of what you did, when [a schedule]
- An attached list of all Concrete Test Cases with an indication of which ones passed and which ones failed (and by default, then, which ones didn't get run)
- Add the Defect Report number for those that failed
- 3 test scripts – you choose which 3 Test Cases, but try to be a little bit original.
- A list of all the defects found (just Titles or *Brief* Descriptions)
- 3 actual Defect Reports
- *Don't* include the Requirements Matrix

Items for your 3 Test Scripts

- Test Case number or ID
- A *brief* description of what the test case is testing – so a developer, for example, could understand what that case is doing
- Procedure/Steps, whether manual or automated. Include any needed setup (so I could run it, for example)
- Expected results

(The requirement being tested is not necessary, since that information is in the Test Plan. Theoretically.)

Items for your 3 Software Defect Reports

- Identifiers: title, number, submitter, submit date
- Program name and version/revision
- Description of the problem
- The Test Case used

Chapter Questions

- Know what should be contained in a test report.
- Know what should be contained in a defect report.
- Know the difference between defect priority and defect severity.

NEXT WEEK:

- Test Oracles (Read Chapter 14 of AO)
- Test Reviews (Read Chapter 3 of GBV)