

The Maintenance Phase - Bug Tracking

Credits:
Carnegie-Mellon
Alan Beccati

Instructor: Peter Baumann

email: pbaumann@constructor.university

tel: -3178

office: room 88, Research 1

Error: Keyboard not attached.
Press F1 to continue.

Harvard Comp. Lab, Mark II log book:

Photo # NH 96566-KN First Computer "Bug", 1945

9/2

9/9

0800 Antam started
 1000 " stopped - antam ✓
 1300 (032) MP-MC ~~1.982147000~~ { 1.2700 9.037847025
 (033) PRO 2 ~~2.130476415~~ } 9.037846995 cond
 cond 2.130476415
 cond 2.130676415

Relays 6-2 in 033 failed special speed test
 in relay " 10.00 test.

Relay 2145
 Relay 3376

1100 Started Cosine Tape (Sine check)
 1525 Started Multi-Adder Test.

1545 Relay #70 Panel F (moth) in relay.



First actual case of bug being found.

1630 Antam started.
 1700 closed down.

Bugs

- **Bug** = inexplicable defect
 - engineering jargon, since 1940s
 - can even trace back to T.A. Edison, 1878: "little faults and difficulties" in an invention
- **Software updates** = new release to
 - cure defects ("bugs")
 - respond to Requests for Enhancement ("RFEs")
 - ...typical incremental release includes both
- Update is either **incremental** ("patch") or **full**
 - Patch particularly critical to ascertain proper interplay of old & new components!

[IETF RFC 9225: Software Defects Considered Harmful](#)



Anatomy of a Bug

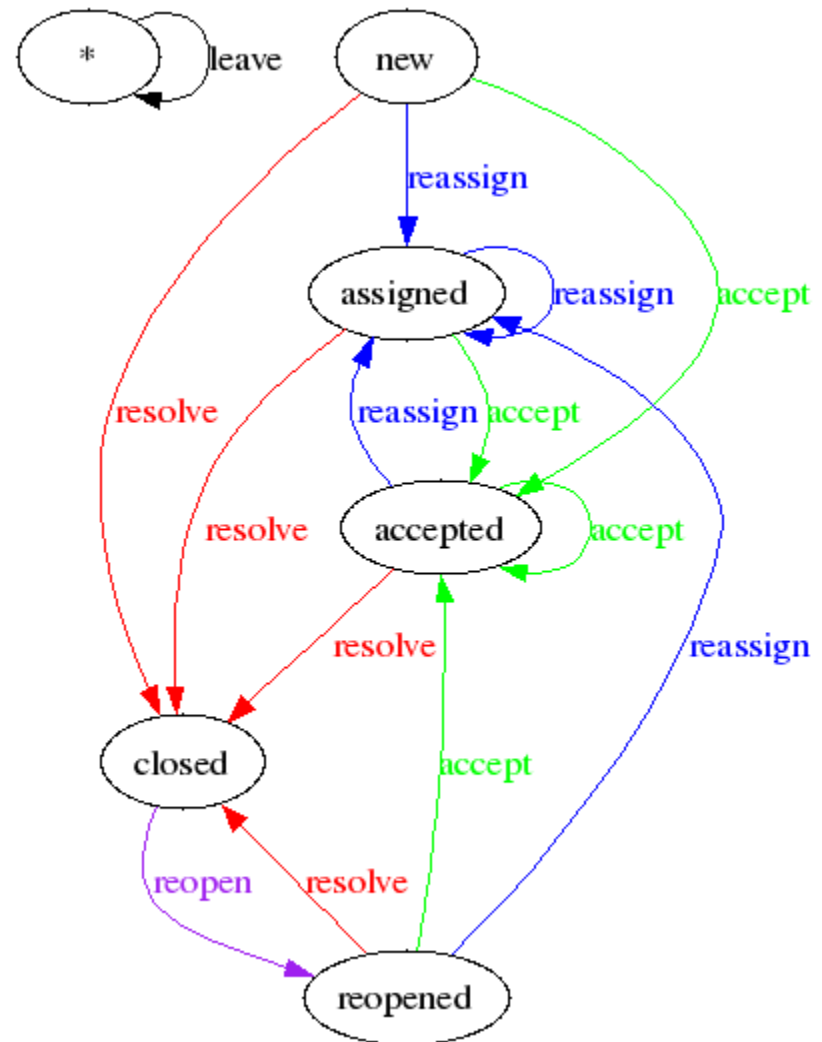
- **Component:** module where the bug occurred
- **Status:** short notes
- **Keywords:** e.g. test cases, help requested
- **Target Milestone:** estimated fix date or release version

Life Cycle of a Bug: Resolution

- Fixed bugs are marked **RESOLVED** and receive a resolution:
 - **FIXED**: source updated and tested
 - **INVALID**: not a bug in the code
 - **WONTFIX**: “feature”, not a bug
 - **DUPLICATE**: already reported elsewhere; include reference
 - **WORKSFORME**: couldn't reproduce

Example: Trac

- Trac = integrated tool for software project management (minimalistic)
- Includes ticket management
 - Ticket types: Bug, task, feature, documentation...
 - Customizable
- open-source project
 - trac.edgewall.org



How to Write Bug Reports

- *“The more effectively a bug is reported, the more likely that an engineer will actually fix it”*
- Bug reports should be:
 - **Reproducible**: provide relevant detail
 - **Specific**: isolate the cause if possible
- Example of a good bug report:
http://bugzilla.mozilla.org/show_bug.cgi?id=2683