JACOBS UNIVERSITY

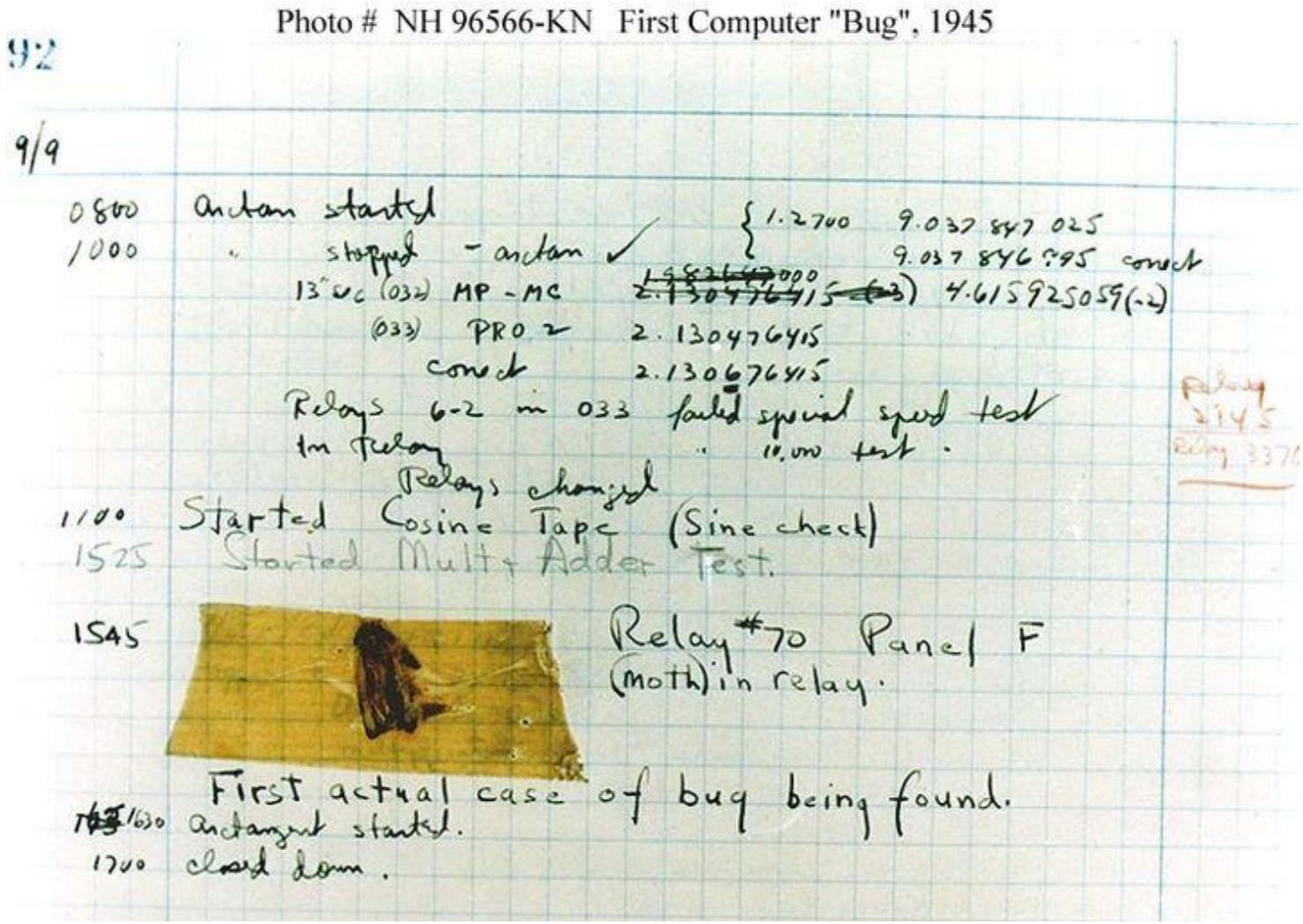
The Maintenance Phase
- Bug Tracking

Credits:
Carnegie-Mellon
Alan Beccati

Instructor: Peter Baumann
email: p.baumann@jacobs-university.de
tel: -3178
office: room 88, Research 1

Error: Keyboard not attached.
Press F1 to continue.

Harvard Comp. Lab, Mark II log book:



Bugs

JACOBS UNIVERSITY

- 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!

Anatomy of a Bug

JACOBS UNIVERSITY

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

Bug Anatomy [2]

JACOBS UNIVERSITY

- Dependency:
  - Bugs that this one depends on
  - Bugs that depend on this one
  - Bugzilla can display the dependency graph
- Attachment:
  - test cases, screen shots, editor logs, patch (diff) file

Life Cycle of a Bug: Start

- Bug log entry is created and assigned to developer with status NEW
- NEW bugs that are idle for a week trigger email reminders
- Reassigning a bug to someone else resets status to (NEW ->) Assigned
- Other fields updated during debugging process (with comments)

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

Life Cycle of a Bug: Check

- QA engineer checks resolved bugs
  - Verifies appropriate actions
  - Mark VERIFIED (okay)
  - Mark REOPENED (found issue)
- After product ships and is accepted by the customer…
  - Mark CLOSED (finally!)

Bugzilla

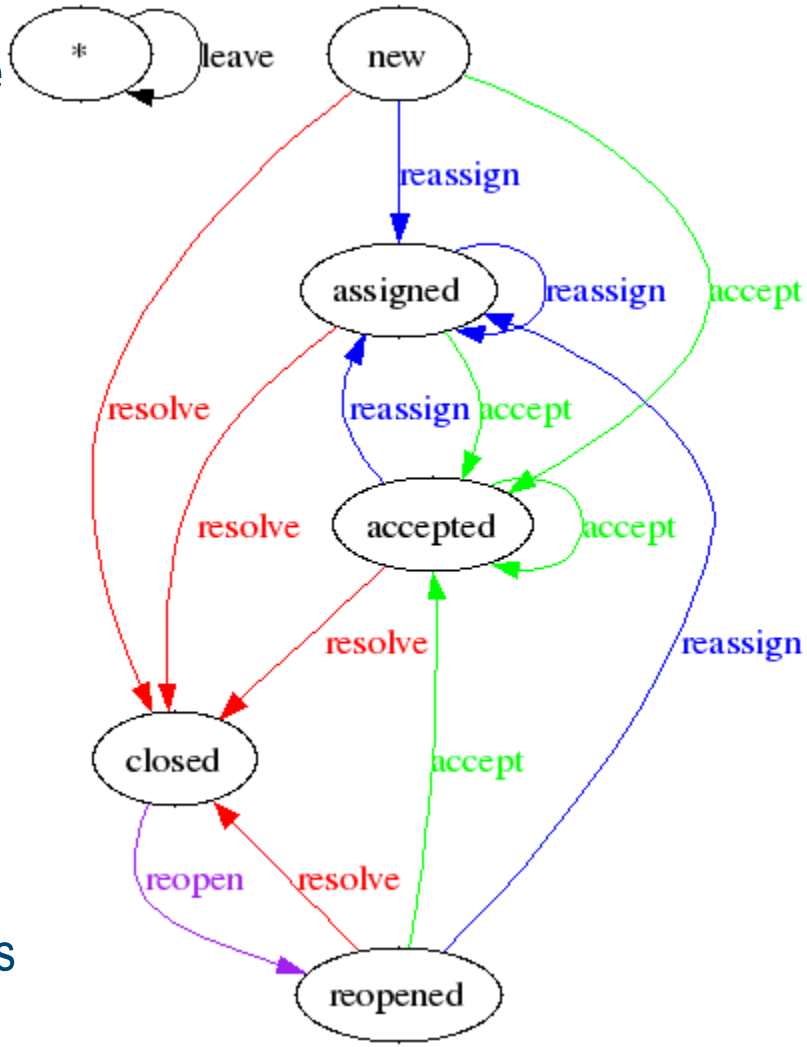
JACOBS UNIVERSITY

- bug tracking system = “a database for bugs”
- Users can:
  - report bugs
  - System will assign them appropriate developers
- Developers can:
  - keep a to-do list
  - prioritize, schedule, and track dependencies between bugs
- Bugzilla = open-source bug tracker
  - bugzilla.mozilla.org/ -- the software
  - www.mozilla.org/bugs/ - the Mozilla bug database (Bugzilla installation)

Trac (bug lifecycle)

JACOBS UNIVERSITY

- Trac is an integrated tool for software project management (minimalistic)
- Includes Ticket management
  - Ticket types: Bug, task, feature, documentation…
  - Completely user-defined
- Ticket workflow (version 0.11)
- open-source project
  - trac.edgewall.org - home page
  - trac.edgewall.org/report/1/ - active tickets



How to Write Bug Reports

JACOBS UNIVERSITY

- "*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
- *…let's give it a try!*

Bug Reports [2]

JACOBS UNIVERSITY

- BAD:
  - "My browser crashed. I think I was on foo.com. I think that this is a really bad problem and you should fix it or else nobody will use your browser."
- GOOD:
  - "I crash each time I go to foo.com (Mozilla build 20000609, Win NT 4.0SP5). This link will crash Mozilla reproducibly unless you remove the "border=0" attribute:

    <IMG SRC="http://foo.com/topicfoos.gif" width=34 height=44 border=0 alt="News">