Defect Reporting and Triage

What's the point of vibe coding if I still have to pay a dev to fix it?

what's the point of vibe coding if at the end of the day i still gotta pay a dev to look at the code anyway. sure it feels kinda cool while i'm typing, like i'm in some flow state or whatever, but when stuff breaks it's just dead weight. i cant vibe my way through debugging, i cant ship anything that actually matters, and then i'm back to square one pulling out my wallet for someone who actually knows what they're doing. makes me think vibe coding is just roleplay for guys who want to feel like hackers without doing the hard part. am i missing something here or is it really just useless once you step outside the fantasy

The Story So Far …

- Quality assurance is critical to software engineering.

- Static (code review, inspection, dataflow analysis) and dynamic (testing, instrumentation) approaches are common

- What happens to all of the bugs those find?

The Very Hungry Caterpillar
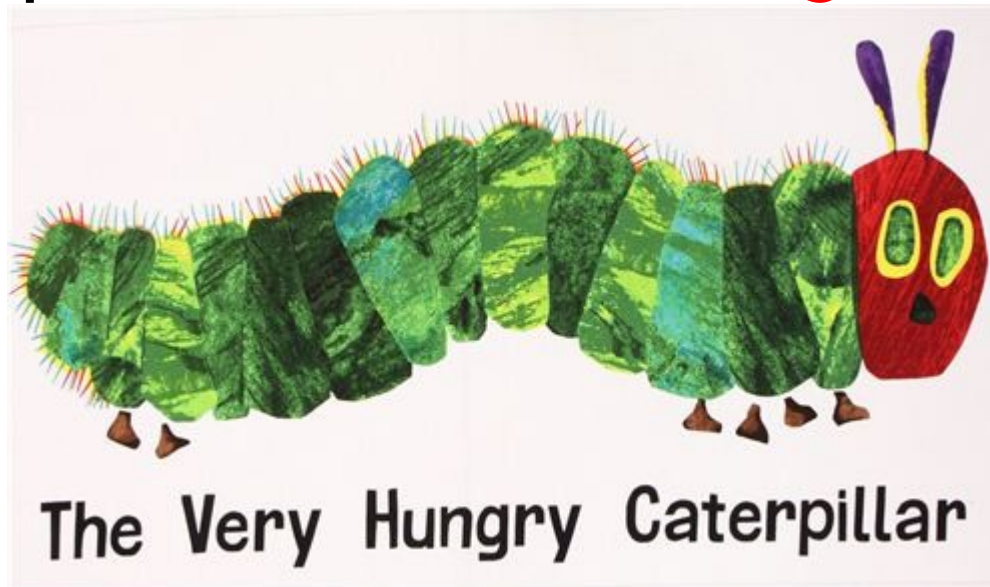
One-Slide Summary

- A software **defect report** includes information and communications related to addressing a software issue.

  - Defect reports have many components.

- Defect reports are subject to **triage** based on **severity** and **priority** information.

- Defect reports have a **lifecycle** that is complicated and non-linear with multiple possible **resolutions**.

Is This Really A Problem?

"Every day, almost 300 bugs appear that need triaging. This is far too much for only the Mozilla programmers to handle."

- Mozilla Developer

Is This Really A Problem?

"Every day, almost 300 bugs appear that need triaging. This is far too much for only the Mozilla programmers to handle."

- Mozilla Developer, *2005*

Plus ça change ...

One More Year ...

100,000/year = 270/day: same rate 15 years in!

Bugzilla — Search Bugs — Browse — Advanced Search — New Account — Log In — Forgot Password

Tue Feb 19 2019 09:53:16 PST

Resolution: --- **Creation date:** (is greater than or equal to) -1d

198 bugs found.

ID	Product	Comp	Assignee	Status	Resolution	Summary	Changed
1528947	Firefox	Firefox Monitor	nobody	NEW	---	Fx monitor runs in private browsing mode	06:45:20
1529020	External Software Af	Other	nobody	NEW	---	Crash in [@ mozilla::ipc::FatalError | mozilla::ipc::IProtocol::HandleFatalError | mozilla::PRemoteDecoderManagerChild::SendPRemoteVideoDecoderConstructor] with scdetour.dll ("Quick Heal" security software)	09:07:12
1528822	Core	XPCOM	nobody	NEW	---	Crash in [@ AsyncShutdownTimeout | profile-before-change | JSON store: writing data]	07:18:16
1528911	Thunderbird	Folder and Message L	nobody	UNCO	---	Pressing Repair folder twice on IMAP inbox deletes all INBOX messages	01:12:04
1528839	Firefox	Untriaged	nobody	UNCO	---	Freeze when using Firefox on Plasma Wayland with wayland support enabled	04:32:47
1528990	bugzilla.mozilla.org	Administration	nobody	NEW	---	Can we rename and alias Web Assembly?	07:25:09
1528802	Cloud Services	Autograph	gguthe	ASSI	---	Deploy Autograph 2.9	Mon 10:02
1528850	Firefox	Untriaged	nobody	UNCO	---	ws-over-hw receive seems unreliable	Mon 15:04
1528928	Firefox	Normandy Server	nobody	NEW	---	There is no error message displayed when trying to publish a "Rollback" experiment that targets an active "Rollout"	02:18:02
1528816	Thunderbird	Message Reader UI	nobody	UNCO	---	View remote content "Preferences" button missing in Conversations message pane	Mon 10:51
1528943	Core	DOM: Events	nobody	UNCO	---	Keyboard events during IME composition should give a special key value	05:35:13
1528900	Firefox Build System	Task Configuration	mh+mozilla	NEW	---	Nightly tasks shouldn't depend on sccache	01:07:23
1528980	Firefox Build System	Try	nobody	NEW	---	|./mach try chooser| should show ip/port the server started on	06:52:13
1528840	Thunderbird	Message Compose Wind	bugzilla2007	ASSI	---	Console error from dispensable code when tabbing forward through recipients in composition; addressingWidgetOverlay.js unused code cleanup	Mon 14:13
1528803	Core	JavaScript Engine	nobody	NEW	---	Handle JSOP_BIGINT in ExpressionDecompiler::decompilePC	05:54:18

Terminology (1/2)

- The software maintenance lexicon is fraught with ambiguity

  - Terms adapted from "standard" engineering, etc.

- A **fault** is an exceptional situation at *run time*

  - In EE: "short circuit", "stuck-at fault"

  - In CS: "trap", "exception"

- A **defect** is any characteristic of a product which hinders its usability for its intended purpose

  - IRL: "design defect", "manufacturing defect"

  - In CS: a **bug** is a *static* defect in the source code

Terminology (2/2)

- A **bug report** (or defect report) provides information about a defect

  - Created by testers, users, tools, etc.

  - Often contains multiple types of information

  - Often tracked in a database

- A **feature request** is a potential change to the intended purpose (requirements) of software

  - In CS: an **issue** is either a bug report or a feature request (cf. "issue tracking system")

- Not used here: "mistake", "error", etc.

These Terms Are Somewhat Subjective
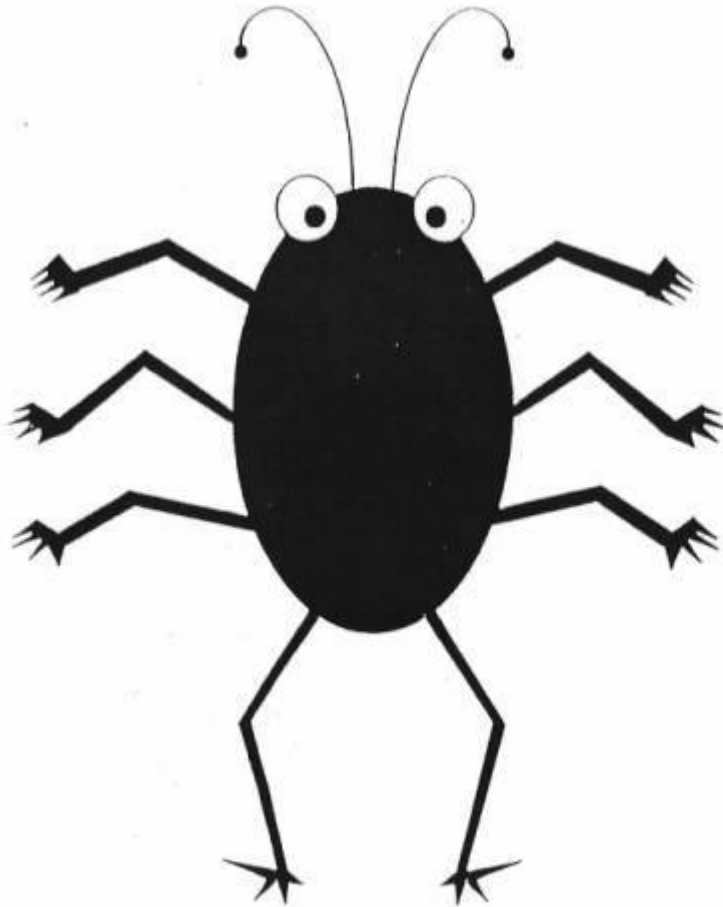
BUG
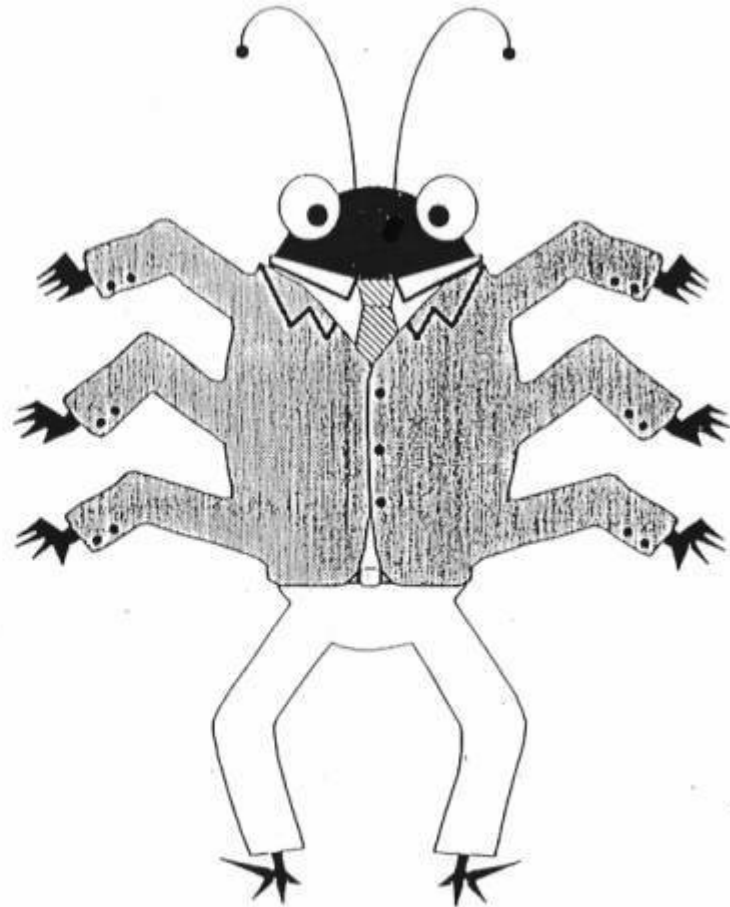
FEATURE

Defect Report Lifecycle

- The **defect report lifecycle** consists of a number of possible stages and actions, including reporting, confirmation, triage, assignment, resolution, and verification.

  - Not every defect report follows the same path

  - The overall process is *not linear*

    - There are multiple entry points, some cycles, and multiple exit points (and some never leave ...)

- The **status** of a defect report tracks its position in the lifecycle ("new", "resolved", etc.)

Report Lifecycle

- Bugzilla is a widely-used open-source issue tracker

- GitHub's built-in issue tracker is similar (less structured)
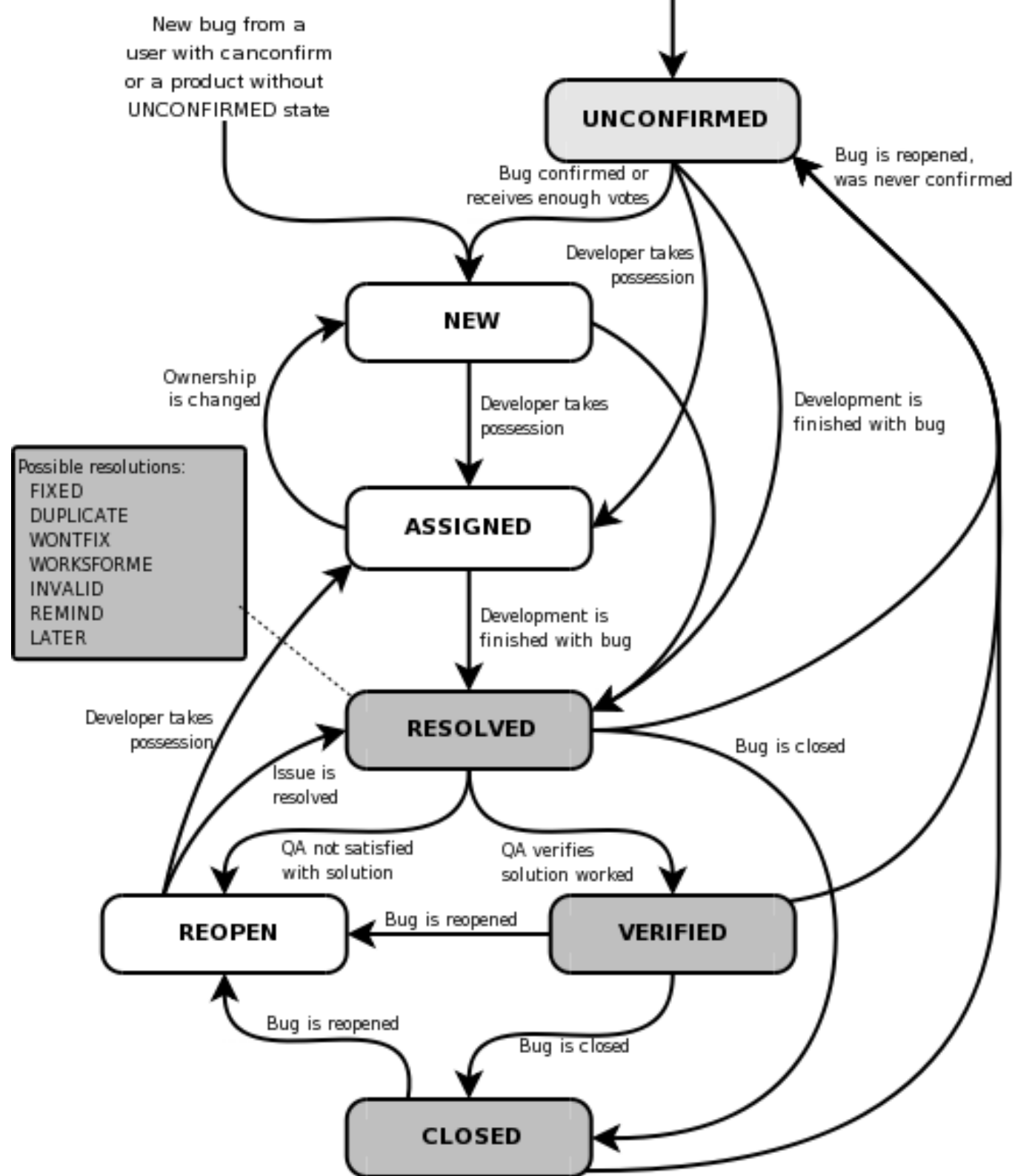
Bug is Reported

- New bug reports enter the system

New bug from a user with canconfirm or a product without UNCONFIRMED state

UNCONFIRMED

Bug is reopened, was never confirmed

Bug confirmed or receives enough votes

Developer takes possession

NEW

Ownership is changed

Developer takes possession

Development is finished with bug

Possible resolutions:
FIXED
DUPLICATE
WONTFIX
WORKSFORME
INVALID
REMIND
LATER

ASSIGNED

Development is finished with bug

Developer takes possession

RESOLVED

Issue is resolved

Bug is closed

QA not satisfied with solution

QA verifies solution worked

REOPEN

Bug is reopened

VERIFIED

Bug is reopened

Bug is closed

CLOSED

Bug Report Sources

- Internal
  - Developers
  - QA / Testers
  - Reports are *usually* detailed, sophisticated
- External
  - Beta Testers
  - End Users
  - Reports are usually more general

Do End Users Submit Bug Reports?

End-User Bug Reports

- Modern view: cannot count on end users to describe bugs in a helpful manner

  - The larger your user base is or the more of a "margin" business model you have, the truer this becomes

- Instead: these are aggregated

Bug Reporting: GitHub

| Issues | Pull requests | Labels | Milestones | | Filters ▾ | 🔍 is:open is:issue | | New issue |

⚠ **104 Open** ✔ 9,660 Closed   Author ▾   Labels ▾   Milestones ▾   Assignee ▾   Sort ▾

⚠ **.form-group-sm .form-group-lg shrink textarea** `confirmed` `css`   💬 4
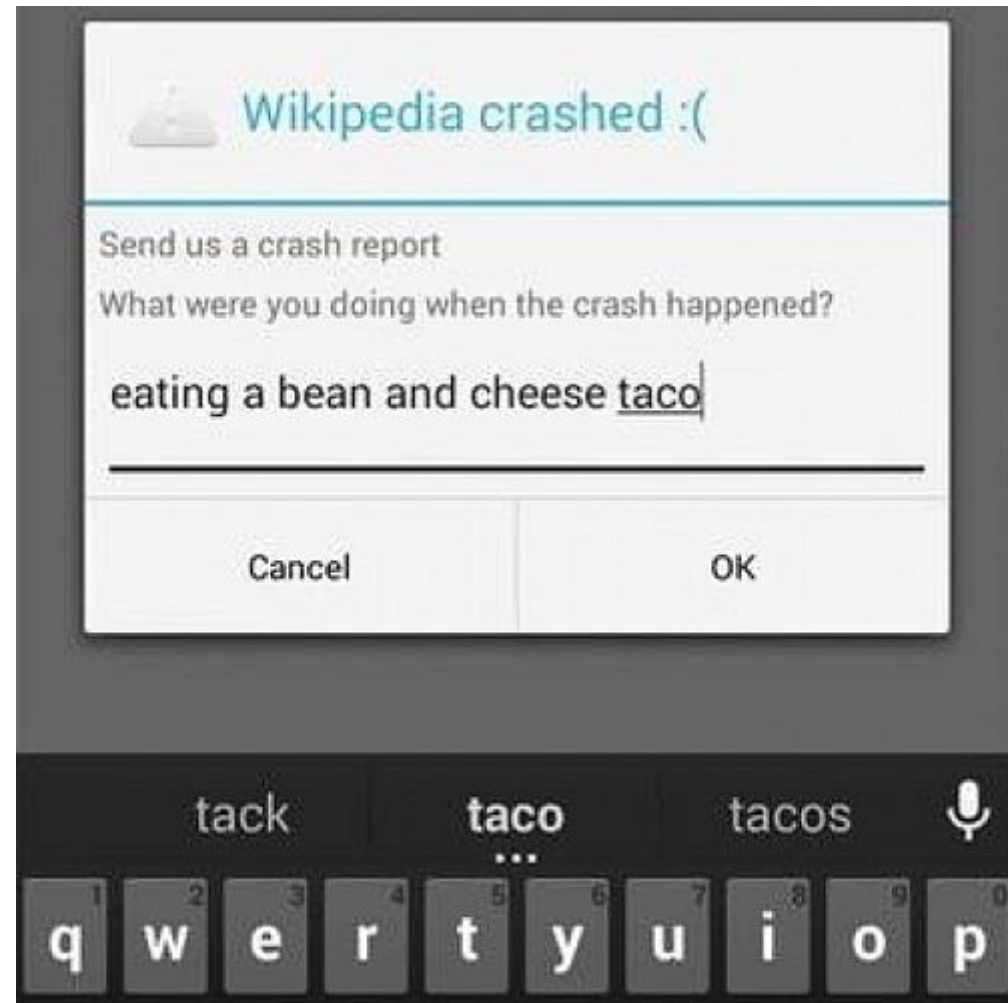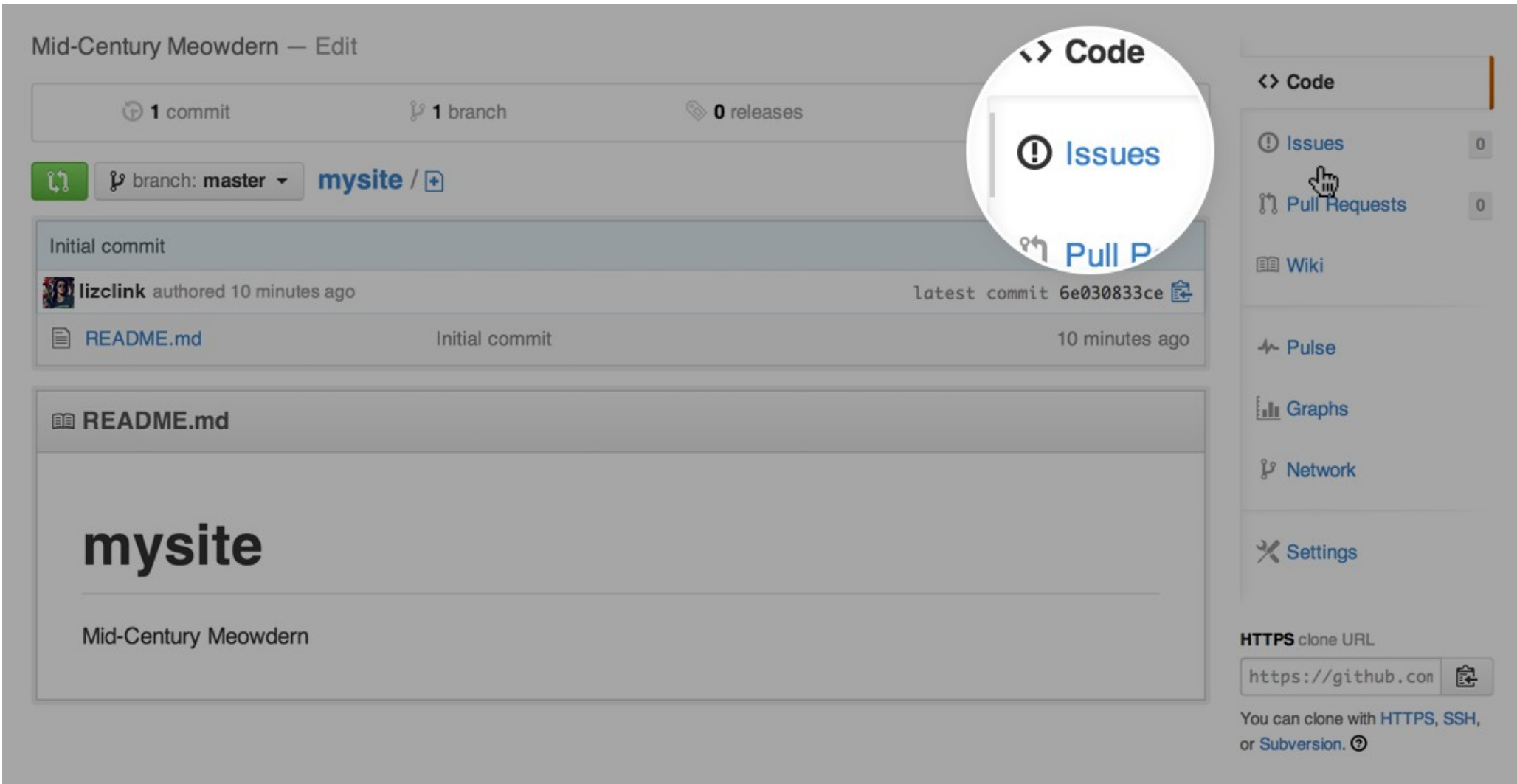#13989 opened 11 hours ago by limitstudios   🚏 v3.2.1

⚠ **Tooltip unnecessarily breaks into multiple lines when positioned to the right** `confirmed` `js`   💬 0
#13987 opened 15 hours ago by hnrch02   🚏 v3.2.1

⚠ **Tooltip Arrows in Modal example facing wrong way** `css`   💬 6
#13981 opened a day ago by SDCore

⚠ **Table improvement** `css`   💬 0
#13978 opened a day ago by Tjoosten

⚠ **docs/dist files** `docs`   💬 7
#13977 opened 2 days ago by XhmikosR   🚏 v3.2.1

⚠ **Potential solution to #4647** `js`   💬 4
#13976 opened 2 days ago by julioarmandof

⚠ **Bootstrap site: right-hand navigation text becomes rasterized after scrolling** `css` `docs`   💬 4
#13974 opened 2 days ago by mg1075   🚏 v3.2.1

⚠ **Dropdown toggle requires two clicks** `js`   💬 1
#13972 opened 2 days ago by Kizmar

19

The Anatomy of a Bug Report

- What should be in a bug report? What is?
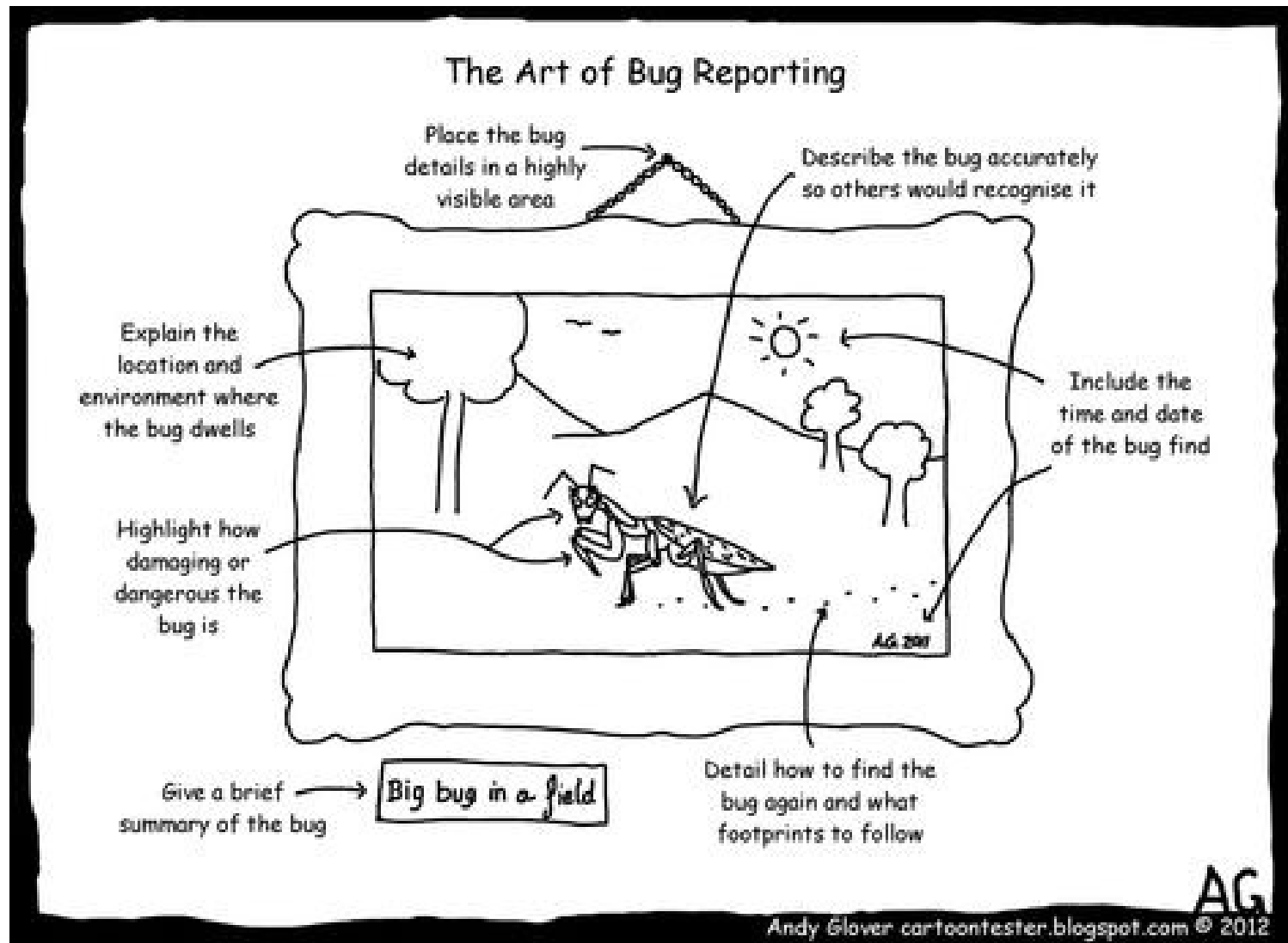
Defect Report Components

Bug 1194147

[AriesKK][Lock Screen]The home button will overlap with the lock icon on lockscreen.

NEW   Assigned to prateekjadhwani

▾ **Status**

			Reported:	3 years ago
Product:	▸ Firefox OS		Modified:	2 years ago
Component:	▸ Gaia::System::Lockscreen			
Status:	**NEW**			

▸ **People** (Reporter: zouyi(Leave from Mozilla), Assigned: prateekjadhwani)

▾ **Tracking**

Version:	**unspecified**
Target:	---
Platform:	**ARM Gonk (Firefox OS)**

▸ **Firefox Tracking Flags** (b2g-v2.5 affected, b2g-master affected)

▾ **Details** (Whiteboard: [2.5-aries-test-run-1])

Whiteboard:	**[2.5-aries-test-run-1]**
Votes:	**0 votes**
QA Whiteboard:	**[MGSEI-Triage+]**

Defect Report Attachments

- Screenshots

- Videos

- Stack Traces

- Data Files

- Note: rarely present

- Note: may come from multiple sources

"Ideal" Defect Report Comment

```
[1.Description]:
[AriesKK v2.5][Lock Screen]Enable PIN, and enter the EMERGENCY CALL from lockscreen, then launch camera by tapping HW-camera
key, and return to EMERGENCY CALL again, tap home button, you will find the home button overlaps with lock icon.
See attachment: AriesKK_v2.5.3gp and logcat.txt
Found at: 10:33

[2.Testing Steps]:
Precondition: Set the lock screen PIN
1. Lock screen;
2. Slide the circle to unlock icon
3. Tap "EMERGENCY CALL"
4. Long tap HW-Camera key to launch camera
5. Tap home button
6. Repeat step 1~step 3
7. Tap home button

[3.Expected Result]:
7. Return to lock screen, and it is shown normally without any overlap

[4.Actual Result]:
7. Home button will overlap with lock icon

[5.Reproduction build]:
Device: Aries KK 2.5(Affected)
Build ID               20150811234258
Gaia Revision          c7875bbc8b32e7b95cc55c9690b03b140905d84d
Gaia Date              2015-08-11 18:18:36
Gecko Revision         https://hg.mozilla.org/mozilla-central/rev/d4f3a8a75577e4af2914a4e899ca2e724f9715c4
Gecko Version          43.0a1
Device Name            aries
Firmware(Release)      4.4.2
Firmware(Incremental)  eng.worker.20150811.230622
Firmware Date          Tue Aug 11 23:06:29 UTC 2015
Bootloader             s1

[6.Reproduction Frequency]:
Always Recurrence,5/5
```

Defect Reports: Conversations

- Defect reports are *not* static

- Instead, they are updated over time

  - Request more info

  - Assign to a dev

  - Discuss solutions

- The report is a log of all relevant activity

- *With your team ...*

Trivia: Entymology

- This group of insects (order Coleoptera) have hardened wing-cases. They make up 40% of all described insects and 25% of all known animal life-forms. They often feed on plants and fungi, break down animal and plant debris, and eat other invertebrates.

Trivia: Music

- This 1967 Beatles effort was their best-selling album in the UK (17x platinum). In addition to the eponymous fictional-group lead track, it includes *With a Little Help from My Friends*, *Lucy in the Sky with Diamonds*, and *When I'm Sixty-Four.* "On release, the album was lauded by the vast majority of critics for its innovations in music production, songwriting and graphic design, for bridging a cultural divide between popular music and legitimate art, and for providing a musical representation of its generation and the contemporary counterculture."

Trivia: Movies

- This giant, lumbering paranormal monster from the Ghostbusters franchise appears when Gozer tells the heroes that it will take the form of the next thing they think of. Ray tries to think of "the most harmless thing ... that could never possibly destroy us."

Psychology: Delayed Gratification

- A child is offered a choice between one reward *now* or two rewards *15 minutes later.*

- Over 600 children took part.

- Some would "cover their eyes with their hands or turn around so that they can't see the tray, others start kicking the desk, or tug on their pigtails, or stroke the marshmallow as if it were a tiny stuffed animal," while others would simply eat the marshmallow as soon as the researchers left.

Psychology: Delayed Gratification

- Results:
  - A minority choose the single reward immediately
  - A majority attempted to wait the 15 minutes
    - One-third of those who attempted succeeded
    - Age was a major correlated factor
    - Trust/belief in reward was also a major factor
- This work is well-known because of the associated follow-up studies and correlations

Psychology: Delayed Gratification

- The ability to delay gratification also correlates with higher SAT scores

- Brain imaging study of a sample from the original Stanford participants when they reached mid-life showed key differences between those with high delay times and those with low delay times in two areas: the prefrontal cortex (more active in high delayers) and the ventral striatum, (more active in low delayers) when they were trying to control their responses to temptations

Psychology: Delayed Gratification

- Also correlates with educational attainment, body-mass index, cognitive and academic competence, and ability to cope with frustration and stress in adolescence

- [Mischel, Walter; Ebbesen, Ebbe B. 1970. "Attention in delay of gratification". Journal of Personality and Social Psychology. 16 (2): 329–337]

- Implications for SE: "quick and dirty" fix or desire to "just start coding" vs. planning and using an SE process

Medical Triage

- Which patients should we address first?

Bug Triage

- Which bugs should we address first?
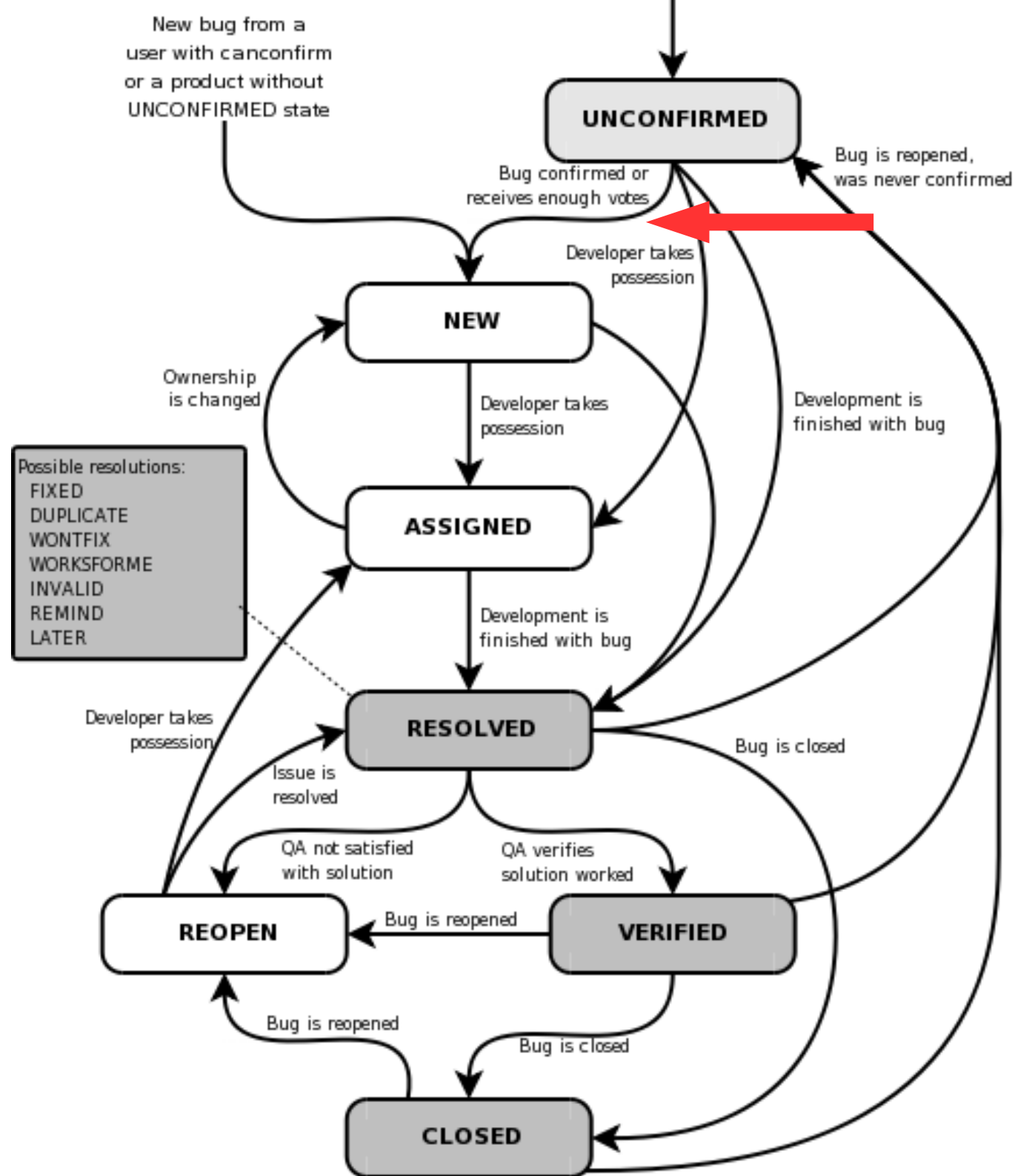
Triage

- **Triage** is the assignment of degrees of urgency to wounds or illnesses to decide the order of treatment of a large number of patients or casualties

- There are always more defect reports than resources available to address them

- Cost-benefit analysis

  - How expensive is it to fix this bug?

  - How expensive is it to *not* fix this bug?

Which Bugs Should We Fix?

- Common Myth:

Severity

- **Severity** is the degree of impact that a defect has on the development or operation of a component or system
  - "cost of not fixing it"
  - Bugzilla severity labels:

Severity	Meaning
Blocker	Blocks further development and/or testing work.
Critical	Crashes, loss of data (internally, not your edit preview!) in a widely used and important component.
Major	Major loss of function in an important area.
Normal	Default/average.
Minor	Minor loss of function, or other problem that does not affect many people or where an easy workaround is present.
Trivial	Cosmetic problem like misspelled words or misaligned text which does not really cause problems.
Enhancement	Request for a new feature or change in functionality for an existing feature.

Priority

- **Defect Priority** (Bug Priority) indicates the importance or urgency of fixing a defect.

- Phabricator examples:

    - **Needs Triage** - Default option, priority has not yet been determined

    - **Unbreak Now!** - Something is broken and needs to be fixed immediately, setting anything else aside

    - **High** - Someone is working or planning to work on this task soon

    - **Normal** - Less than High, but someone still plans to work on it

    - **Low** - Less than Normal, but someone still plans to work on it

    - **Lowest** - Nobody plans to work on this task

Priority Assignment Example

- Phabricator Agile example:

  - **High priority** for tasks committed for the current sprint, or that need to find an owner who can start working on them soon

  - **Normal priority** for tasks that are not critical for the current sprint or candidates for a next sprint

  - **Low priority** for tasks that we can live without, usually sitting in the backlog, sometimes added to a sprint

  - "As a rule of thumb, limit High priority task assignments for a single person to three, five in exceptional times."

Severity vs. Priority

- Severity and Priority are often correlated, but are officially independent

    - A "Normal" Severity issue could currently be "Low" Priority if there are many outstanding "Critical" Severity issues, etc.

- Severity and Priority are used together (along with complexity, risk, etc.) to evaluate, prioritize and assign the resolution of reports

    - Distributed on-line voting (e.g., in open source)

    - In-person meeting (e.g., commercial)

Simple Lie

- Supply + Demand → Price
- Severity + Priority → Triage

Bug Assignment

- Who should fix this bug?
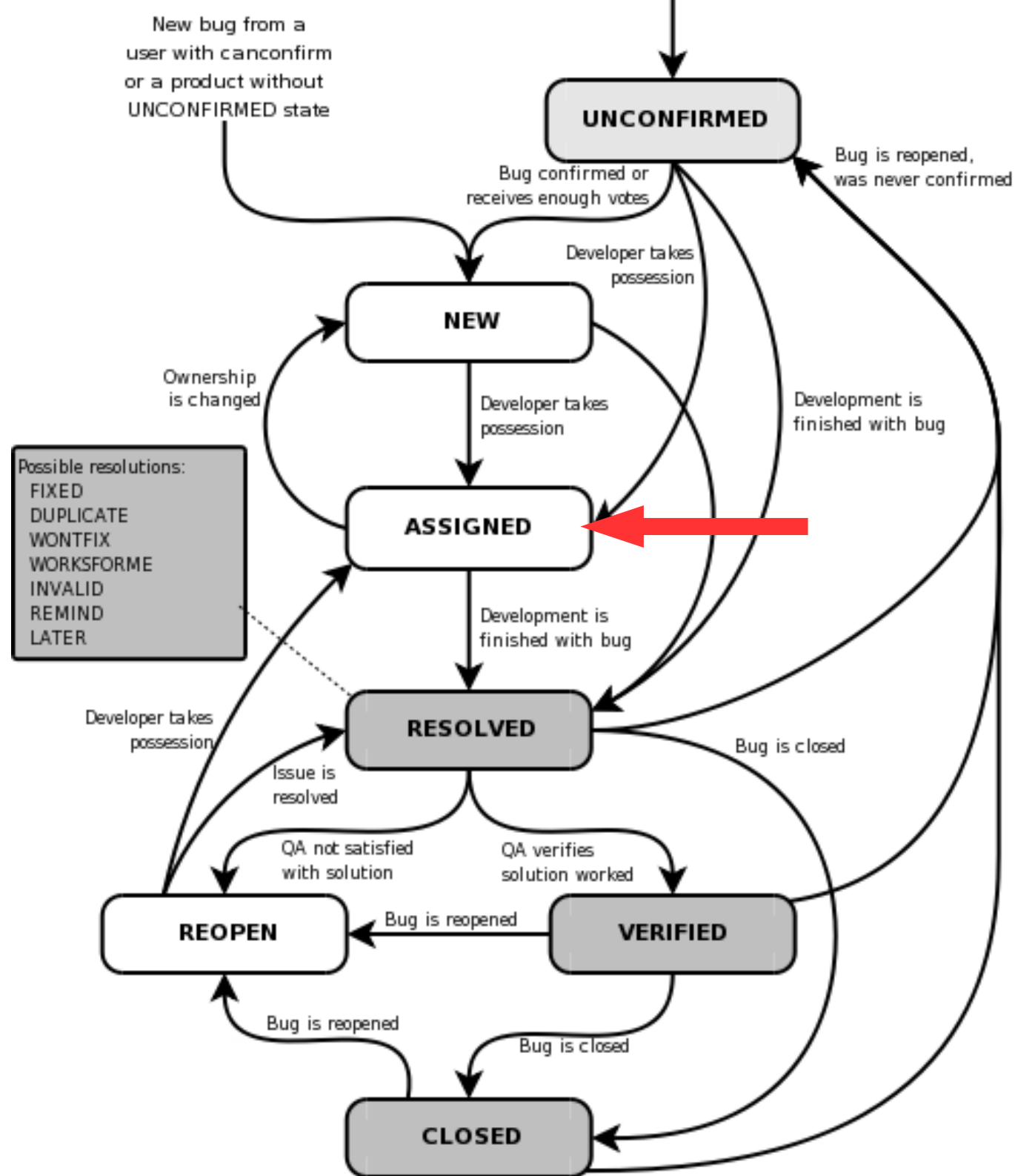
Example

- **Severity** and **Priority** discussions

- **Assignment** discussions

Bug 1383052

**Intermittent dom/canvas/test/test_bitmapre[...]
Host::Lock]**

NEW Unassigned

▼ **Status** (NEW bug which is in the backlog of work)

Product:	▸ Core
Component:	▸ Canvas: 2D
Importance:	P3 critical
Status:	NEW

Gerry Chang [:gchang]
Comment 2 · 3 years ago

```
This issue still happens in below build.
Build ID            20150825021113
Gaia Revision       b441bde54293bea5254dc340845effe951fa3906
Gaia Date           2015-08-24 17:47:19
Gecko Revision      https://hg.mozilla.org/mozilla-central/rev/04b8c412d9f58fb
Gecko Version       43.0a1
Device Name         aries
Firmware(Release)   4.4.2

Hi Fred,
Can you help to dispatch this to right owner?
```

Flags: needinfo?(gasolin)

gasolin@mozilla.com
Comment 3 · 3 years ago

```
Though its not a priority with soft homescreen button,

greg could you put it in the backlog?
```

Flags: needinfo?(gasolin) → needinfo?(gweng)

Greg Weng [:snowmantw][:gweng][:λ]
Comment 4 · 3 years ago

```
Could we analyze if it's Software Home Button issue or LockScreen issue first?
```

Flags: needinfo?(gweng)

Prateek Jadhwani [:prateekjadhwani]   (Assignee)
Comment 5 · 2 years ago

```
I will look into it.
```

Flags: needinfo?(administration)

Prateek Jadhwani [:prateekjadhwani]   (Assignee)
Updated · 2 years ago

Flags: needinfo?(administration)

Prateek Jadhwani [:prateekjadhwani]   (Assignee)
Comment 6 · 2 years ago

```
:gweng Would it be possible for you to assign it to me?
Thanks
```

Flags: needinfo?(gweng)

[:fabrice] Fabrice Desré
Updated · 2 years ago

Assignee: nobody → prateekjadhwani

Defect Assignment

- An **assignment** associates a developer with the responsibility of addressing a defect report

- The current state of the art is "manual"

  - Distributed: developers watch the incoming bug report queue and claim defects for themselves

  - Centralized: one or more people in QA watch the incoming bug report queue and assign reports to a pool of developers

- Usually based who "owns" the implicated code

Near Future: Automatic Assignment

Who Should Fix This Bug?

John Anvik, Lyndon Hiew and Gail C. Murphy
Department of Computer Science
University of British Columbia
{janvik, lyndonh, murphy}@cs.ubc.ca

ABSTRACT

Open source development projects typically support an open bug repository to which both developers and users can report bugs. The reports that appear in this repository must be triaged to determine if the report is one which requires attention and if it is, which developer will be assigned the responsibility of resolving the report. Large open source developments are burdened by the rate at which new bug reports appear in the bug repository. In this paper, we present a semi-automated approach intended to ease one part of this process, the assignment of reports to a developer. Our approach applies a machine learning algorithm to the open bug repository to learn the kinds of reports each developer resolves. When a new report arrives, the classifier produced by the machine learning technique suggests a small number of developers suitable to resolve the report. With this approach, we have reached precision levels of 57% and 64% on the Eclipse and Firefox development projects respectively.

However, this potential advantage also comes with a significant cost. Each bug that is reported must be *triaged* to determine if it describes a meaningful new problem or enhancement, and if it does, it must be assigned to an appropriate developer for further handling [13]. Consider the case of the Eclipse open source project[1] over a four month period (January 1, 2005 to April 30, 2005) when 3426 reports were filed, averaging 29 reports per day. Assuming that a triager takes approximately five minutes to read and handle each report, two person-hours per day is being spent on this activity. If all of these reports led to improvements in the code, this might be an acceptable cost to the project. However, since many of the reports are duplicates of existing reports or are not valid reports, much of this work does not improve the product. For instance, of the 3426 reports for Eclipse, 1190 (36%) were marked either as invalid, a duplicate, a bug that could not be replicated, or one that will not be fixed.

As a means of reducing the time spent triaging, we present

Near Future: Automatic Assignment

Who Should Fix This Bug?

Abstract—Large software development projects receive many bug reports and each of these reports needs to be triaged. An important step in the triage process is the assignment of the report to a developer. Most previous efforts towards improving bug report assignment have focused on using an activity-based approach. We address some of the limitations of activity-based approaches by proposing a two-phased location-based approach where bug report assignment recommendations are based on the predicted location of the bug. The proposed approach utilizes a noun extraction process on several information sources to determine bug location information and a simple term weighting scheme to provide a bug report assignment recommendation. We found that by using a location-based approach, we achieved an accuracy of 89.41% and 59.76% when recommending five developers for the Eclipse and Mozilla projects, respectively.

Index Terms—Bug Report Assignment, File Activity Histories, Named Entity Recognition, POS Filtering, Mining Software Artifacts.

Seven Years Later

ABSTRACT

Open source developm
bug reposito to wh
port bugs. The re
be triaged to determ
attention and if it is,
responsibility of resol
velopments are burde
ports appear in the bu
a semi-automated app
process, the assignment of reports to a developer. Our approach applies a machine learning algorithm to the open bug repository to learn the kinds of reports each developer resolves. When a new report arrives, the classifier produced by the machine learning technique suggests a small number of developers suitable to resolve the report. With this approach, we have reached precision levels of 57% and 64% on the Eclipse and Firefox development projects respectively.

age also comes with a sig-
reported must be *triaged*
eaningful new problem or
must be assigned to an ap-
andling [13]. Consider the
roject[1] over a four month
1 30, 2005) when 3426 re-
eports per day. Assuming
y five minutes to read and
ours per day is being spent
ports led to improvements
in the code, this might be an acceptable cost to the project. However, since many of the reports are duplicates of existing reports or are not valid reports, much of this work does not improve the product. For instance, of the 3426 reports for Eclipse, 1190 (36%) were marked either as invalid, a duplicate, a bug that could not be replicated, or one that will not be fixed.
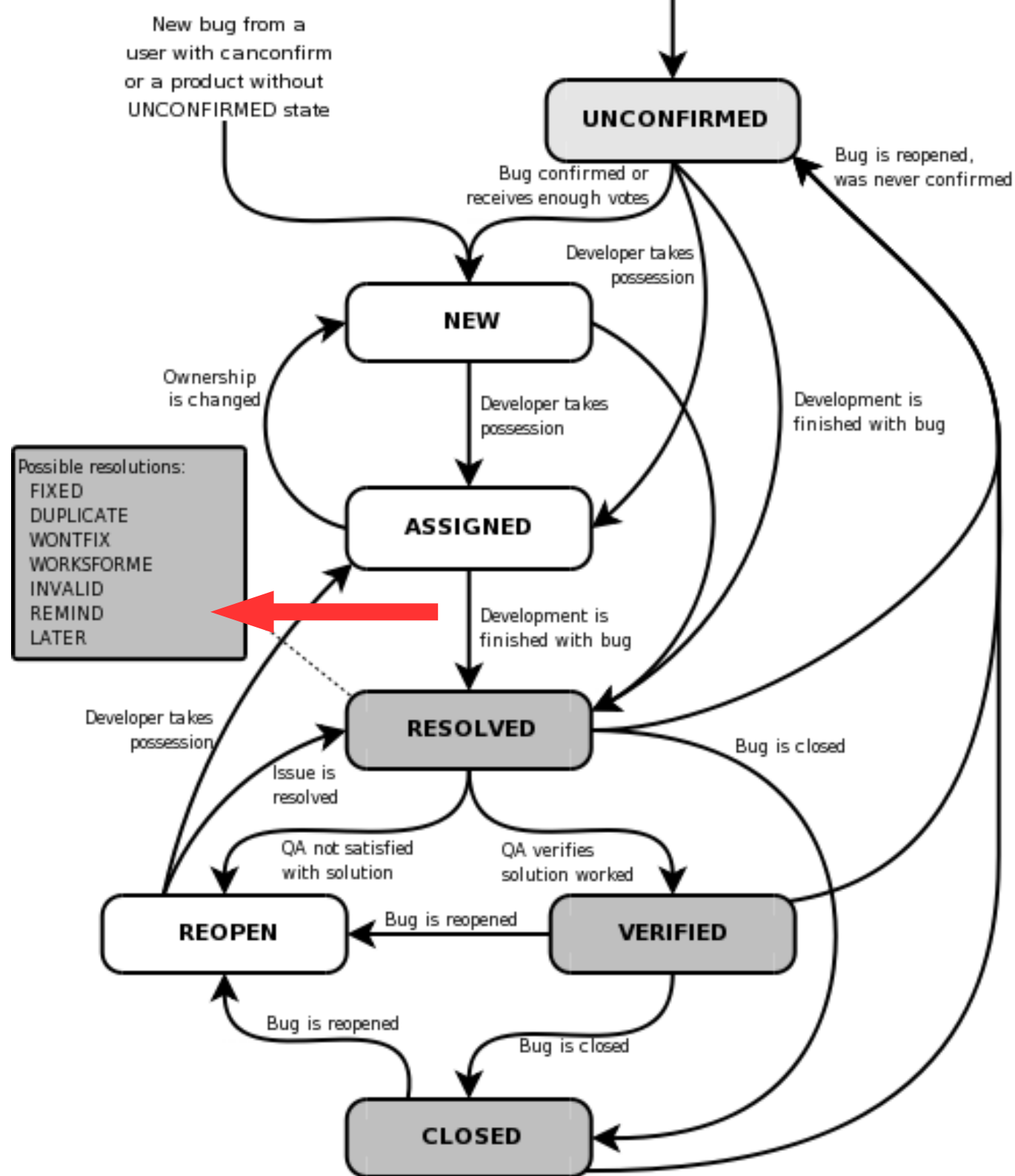
As a means of reducing the time spent triaging, we present
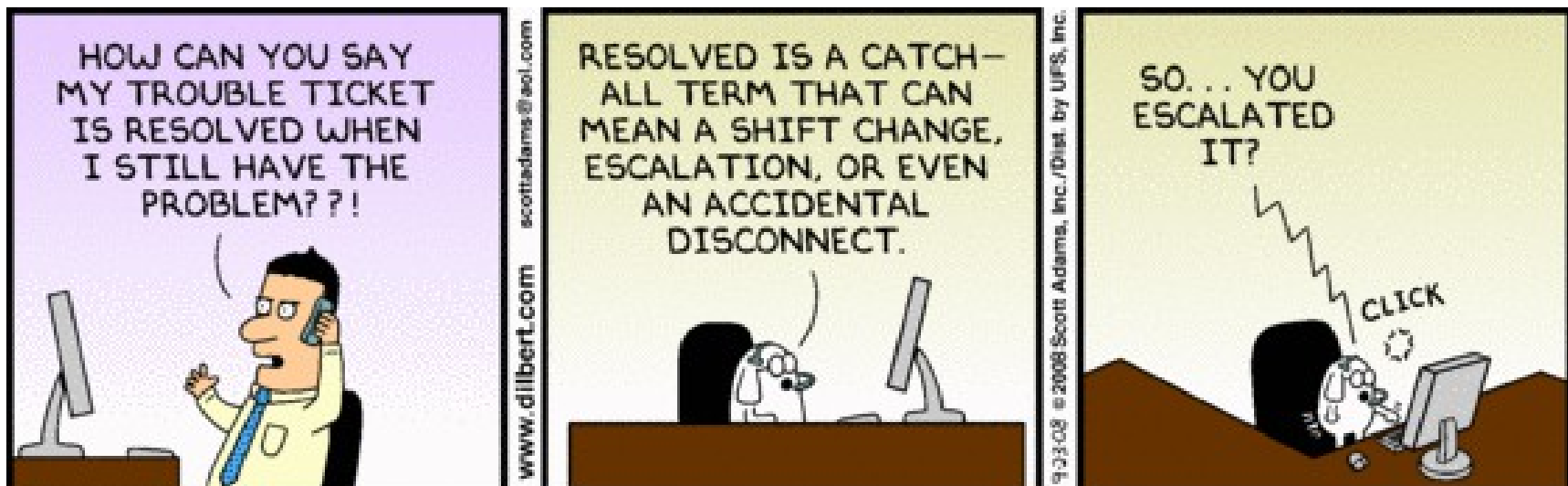
Bug Resolution

- Did we fix it?

Defect Resolution

- Now that the defect report has been assigned to a developer, it can be localized, debugged, etc. Those are future lecture topics!

- A defect report **resolution** status indicates the result of the most recent attempt to address it

  - Important: resolved need not mean "fixed"

Possible Resolutions

- Bugzilla resolution options:
    - **FIXED** (give commit #)
    - **INVALID** (bug report is invalid)
    - **WONTFIX** (we don't ever plan to fix it)
    - **DUPLICATE** (link to other bug report #)
    - **WORKSFORME** (cannot reproduce, a.k.a. "WFM")
    - **MOVED** (give link: filed with wrong project)
    - **NOTABUG** (report describes expected behavior)
    - **NOTOURBUG** (is a bug, but not with our software)
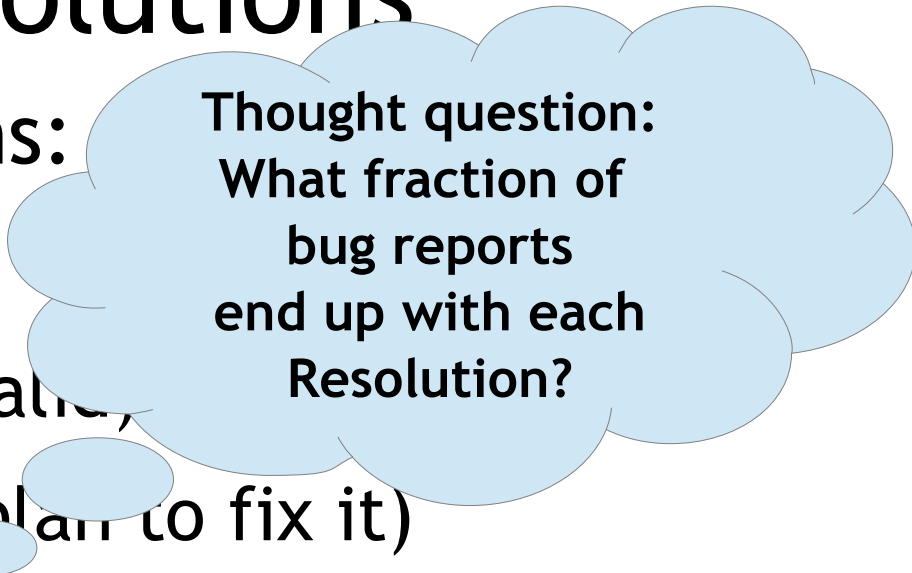   - **INSUFFICIENTDATA** (cannot triage/fix w/o more)

Possible Resolutions

- Bugzilla resolution options:

  - **FIXED** (give commit #)

  - **INVALID** (bug report is invalid)

  - **WONTFIX** (we don't ever plan to fix it)

  - **DUPLICATE** (link to other bug report #)

  - **WORKSFORME** (cannot reproduce, a.k.a. "WFM")

  - **MOVED** (give link: filed with wrong project)

  - **NOTABUG** (report describes expected behavior)

  - **NOTOURBUG** (is a bug, but not with our software)

  - **INSUFFICIENTDATA** (cannot triage/fix w/o more)

Thought question: What fraction of bug reports end up with each Resolution?

Duplicate, Invalid

A significant fraction of submitted bug reports are spurious duplicates that describe already-reported defects. Previous studies report that as many as 36% of bug reports were duplicates or otherwise invalid [2]. Of the 29,000 bug reports used in the experiments in this paper, 25.9% were identified as duplicates by the project developers.

[Jalbert et al. Automated Duplicate Detection for Bug Tracking Systems. DSN 2008.]

Reopen?

- I thought we fixed it!
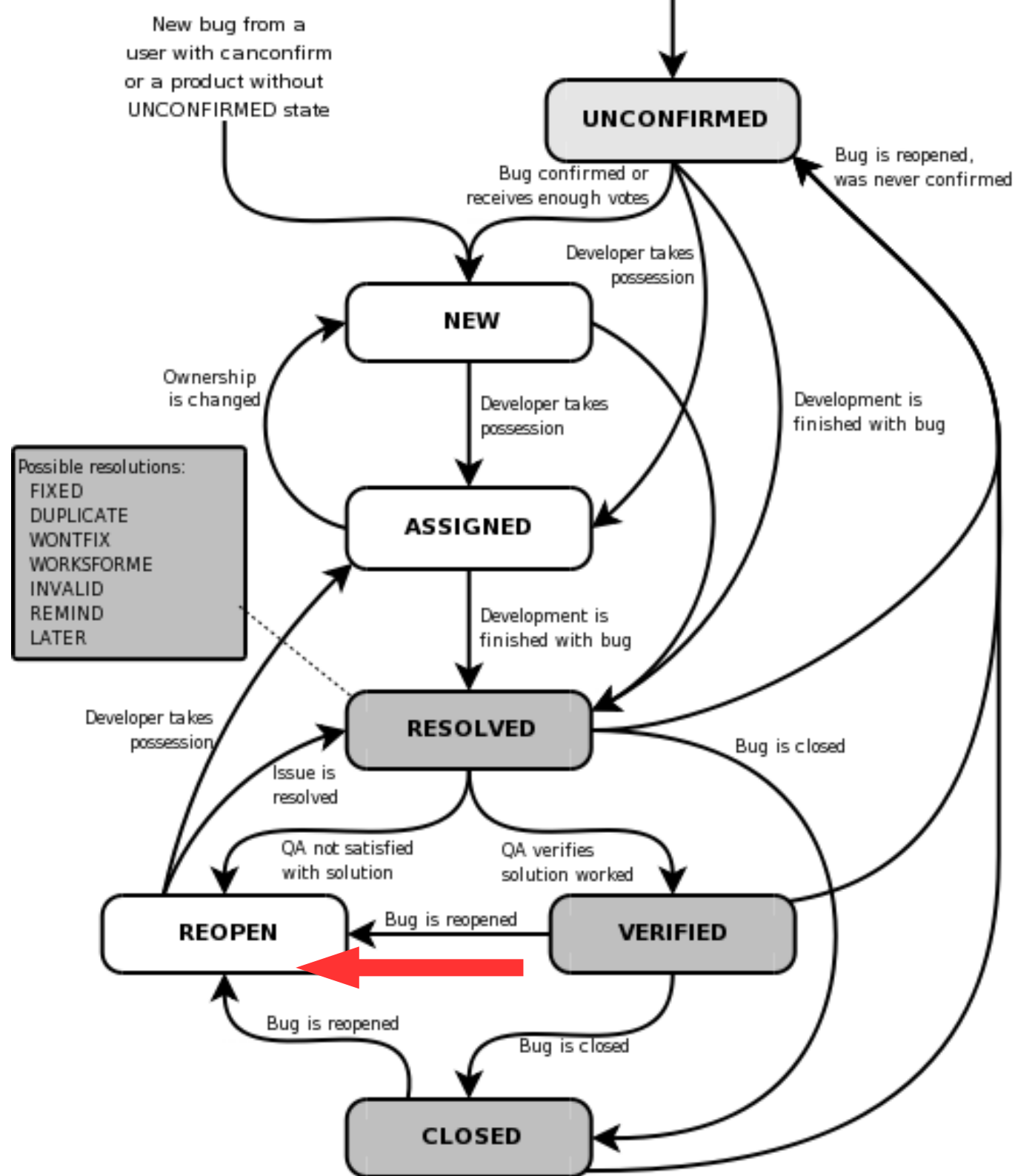
Reopened

- A defect report that was previously resolved (e.g. "FIXED") may be **reopened** if later evidence suggests the old resolution is no longer adequate

  - "We thought this fixed it, but now others are reporting it."

  - "We thought this was out of scope, but now we really need to address it."

- Compare: regression testing

- Surely this only happens rarely?

Many Fixes Are Wrong
Even On Mature, Critical Software

This paper presents a comprehensive characteristic study on incorrect bug-fixes from large operating system code bases including Linux, OpenSolaris, FreeBSD and also a mature *commercial* OS developed and evolved over the last *12* years, investigating not only the mistake patterns during bug-fixing but also the possible *human reasons* in the development process when these incorrect bug-fixes were introduced. Our major findings include: (1) at least 14.8%~24.4% of sampled fixes for post-release bugs [1] in these large OSes are incorrect and have made impacts to end users. (2) Among several common bug types, concurrency bugs are the most difficult to fix correctly: 39% of concurrency bug fixes are incorrect. (3) Developers and reviewers for incorrect fixes usually do not have enough knowledge about the involved code. For example, 27% of the incorrect fixes are made by developers who have never touched the source code files associated with the fix. Our results provide useful guidelines to design new tools and also to improve the development process. Based on our findings, the commercial software

[Yin et al. How Do Fixes Become Bugs? ESEC/FSE 2011.]

Bug Report Characteristics

Large Study of Bug Reports

- 2000 defect reports in Linux, Mozilla, Apache

- Memory Bugs: ~15%; Semantic Bugs: ~75%; Concurrency Bugs: ~10%

- Bug→Crash: ~20%; Bug→Wrong Behavior: ~80%

  - Why Crash? Memory Bugs ~55%

- Most common? ~50% of Mozilla bugs are GUI issues

- Whence security bugs? 30% memory bug causes (severe), 70% semantic bug causes

[Tan et al. Bug Characteristics in Open Source Software. EMSE 2014.]

Questions?

- Exam #1 looms ... (plus HW #3)

*Python Installation Was Successful ✅

for the last time, simply closing a bug report doesn't fix the bug