

# Bug reporting, tracking

- Bugs happen, especially in complex software
- Generally treated as a bug anytime after dev signs off on the code, basically saying “this code is fine”
- May be found anytime in the lifespan of the code – soon or long long after release (maybe some other change in the code or the way it is used reveals the bug years later)
- With lots of code, lots of devs, and lots of bugs, we need a way to keep track of them all

# Bug reports

- Sooner or later, someone observes behaviour that they think is a bug: they need some way to tell the maintainers of the code that something seems broken
- Reports may come in via email, text, chat, phone, etc
- Dev who works on the code may never get to speak directly to the person reporting the bug: need to gather enough info for the dev to work with in tracking down/fixing the problem
- Different organizations have different tools and levels of formality for reporting bugs

# Good bug reports

- Give dev the minimal sequence of steps needed to reproduce the bug
- Tell dev what they expected (should) happen
- Tell dev what actually happened, i.e. why this is a bug
- Give dev an idea of how critical/severe the problem is for reporter (e.g. trivial, minor, significant, serious, urgent)
- Identify which version of the product, and what platform/environment it's being run in (e.g. v3.1 on Win10)

# What happens next?

- Again, different organizations have different levels of formality, but someone needs to review bug report
- Decide if it's a new bug or one we already knew about
- Decide if it's a bug at all, or if user misunderstood what was supposed to happen
- Get preliminary guesstimate of how difficult it will be to track down the bug and (potentially) how difficult it may be to fix
- Someone needs to decide yes/no, will we try to fix it, who it will be assigned to, and who will be paying for it

# Keeping track of the bugs

- Many organizations use bug tracking software/databases, so it's easy to look up old/current/new bugs and their current status
- See wikipedia's comparison of issue-tracking systems [en.wikipedia.org/wiki/Comparison\\_of\\_issue-tracking\\_systems](https://en.wikipedia.org/wiki/Comparison_of_issue-tracking_systems)
- Widely different formats for entering and searching the databases, and different levels of detail for amount/type of data stored about each bug (status, dates reported, analyzed, fixed, nature of bug, date/build introduced, etc)

# Analyzing bug data

- The obvious question: what are the current known bugs
- Many other questions we may be able to answer
- Are there trends in when bugs were introduced (specs, design, implementation, maintenance, versions...)
- Are there trends in type/area bugs appear in (network code, database code, user interface code, etc)
- Are there trends in how much time/money we're spending in fixing bugs
- Are there trends in how long it takes us to find/fix bugs