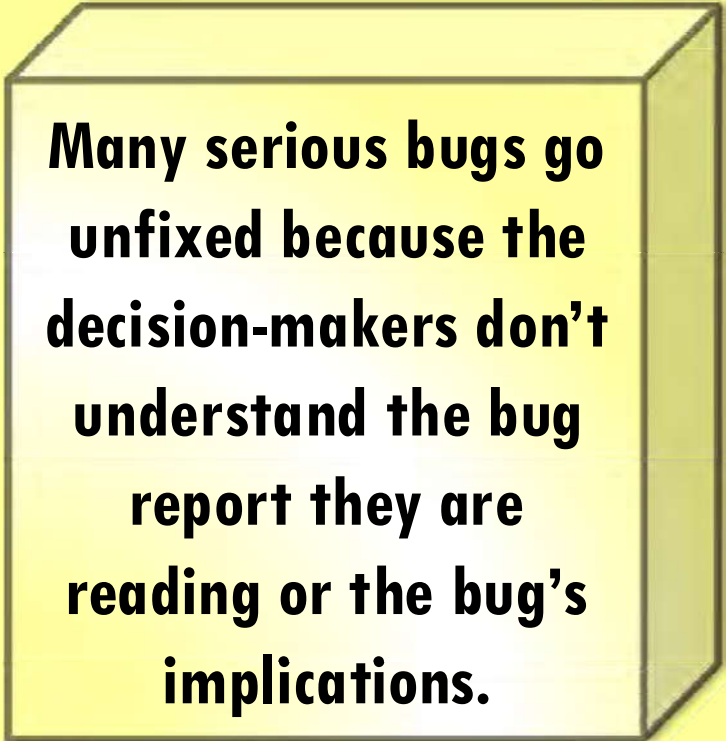




Reviewing Part 1

Part I was about SCOPE: **What should we report?**

- A bug is anything that threatens value to a favored stakeholder.
- A bug report is an assertion that a product could be better than it is.
- The idea of “better” is subjective—what’s better for one stakeholder might be worse for another.
- A bug report is **justified** if it exposes a problem that does in fact reduce value for a stakeholder with influence.



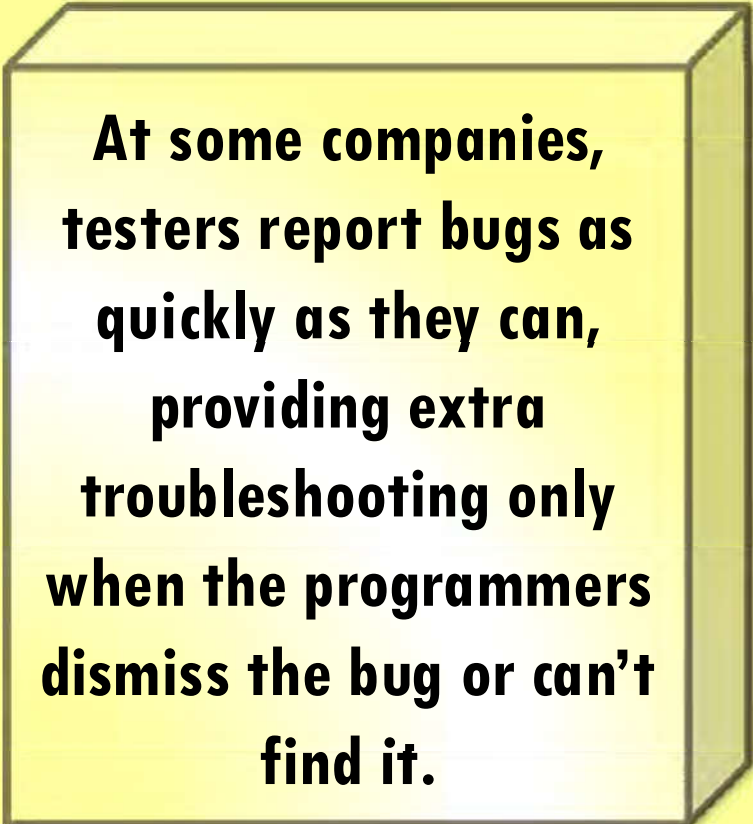
Many serious bugs go unfixed because the decision-makers don't understand the bug report they are reading or the bug's implications.

It's not only about reporting the bug

- Client experienced a wave of serious product recalls (defective firmware)
 - Why were these serious bugs not found in testing?
 - *They WERE found in testing AND reported*
 - Why didn't the programmers fix them?
 - *They didn't understand what they were reading*
 - What was wrong with the bug reports?
 - *The problem is that the testers focused on creating reproducible failures, rather than on the quality of their communication.*
- Looking over 5 years of bug reports, I could predict fixes better by clarity/style/attitude of report than from severity

But there are tradeoffs

1. The more time you spend on each bug, the fewer bugs you have time to find and report
2. If you spend lots of troubleshooting time getting more information for the programmers, how much time does this save them?
 - *If an hour of your investigative time saves the programmer 10 minutes, is this a cost-effective allocation of resources?*



At some companies, testers report bugs as quickly as they can, providing extra troubleshooting only when the programmers dismiss the bug or can't find it.

22

Bug Advocacy = Selling Bugs.

Time is in short supply.

People are overcommitted.

If you want someone to fix your bug, you have to make them want to do it.

- **Your** bug? (Someone else made the bug, but once you find it, it's yours too.)

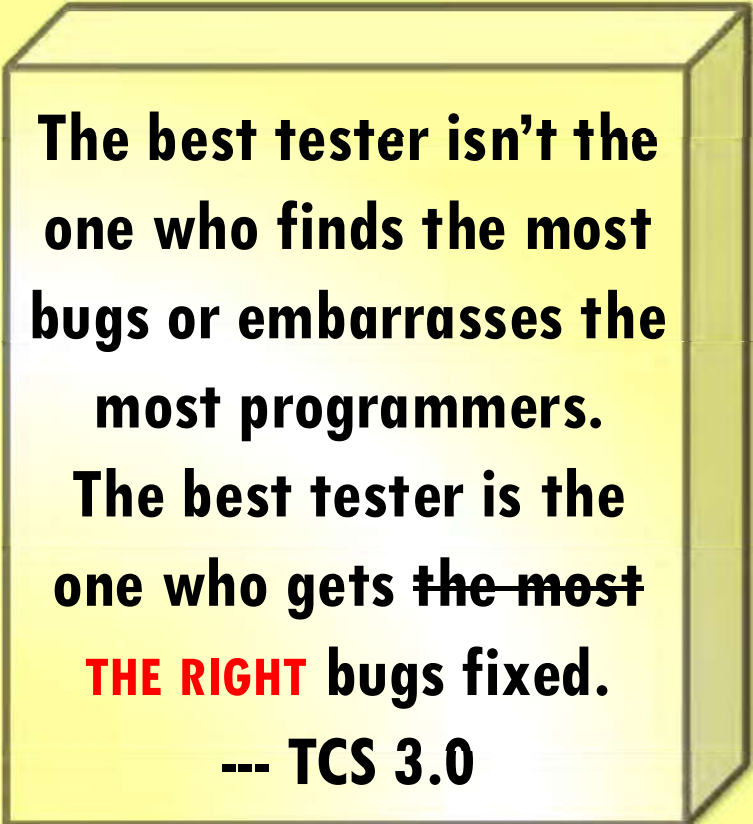
The art of motivating someone to do something that you want them to do is called *sales*.

Your task is to communicate effectively with human decision-makers. Purely technical cost/benefit tradeoffs miss the bigger picture.

Bug Advocacy

It's not just about reporting bugs.

- It's about presenting a bug in its strongest (honestly described) light.
- It's about presenting a bug in a way that connects with the concerns of stakeholders with influence—and if one particular stakeholder will be most affected, by making sure she gets the message clearly.
- It's about communicating so well that your report enables good decision-making.



The best tester isn't the one who finds the most bugs or embarrasses the most programmers. The best tester is the one who gets ~~the most~~ **THE RIGHT bugs fixed.**
--- TCS 3.0

How to sell bugs

Sales revolves around two fundamental objectives:

- **Motivate the buyer**
 - Make her WANT to fix the bug.
- **Overcome objections**
 - Get past her reasons and excuses for not fixing the bug.



**Today we focus on
writing a motivating
report.**

**Later, we consider the
objections**

Motivating the bug fixer

Some programmers want to fix a bug if:

- It looks really bad.
- It will affect lots of people.
- Getting to it is trivially easy.
- It is a breach of contract. ✓
- A bug like it has embarrassed the company, or a competitor.
- It looks like an interesting puzzle and piques the programmer's curiosity.
- Management (that is, someone with influence) has said that they really want it fixed.
- You said you want this particular bug fixed, and the programmer likes you, trusts your judgment, is susceptible to flattery from you, or owes you a favor.



The bug tracking system

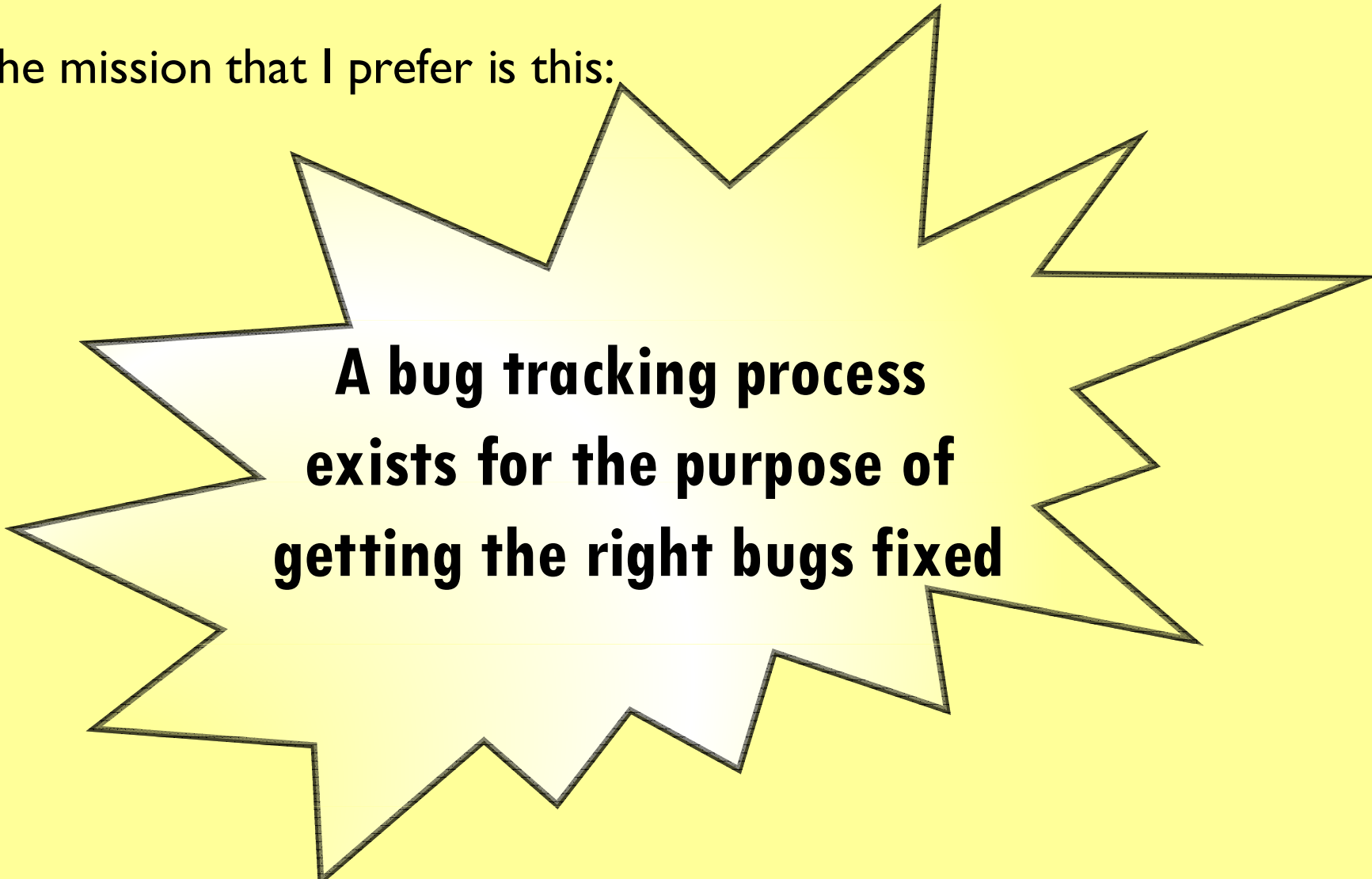
Testers report bugs into a bug tracking system.

- The system (implicit culture or explicit policies, procedures, mission) determine such things as:
 - what types of problems they submit
 - what details go into the reports
 - who has access to the data, for what purpose
 - what summary reports and statistics are available

JIRA (Atlassian)
Github Issues

Mission of bug tracking systems

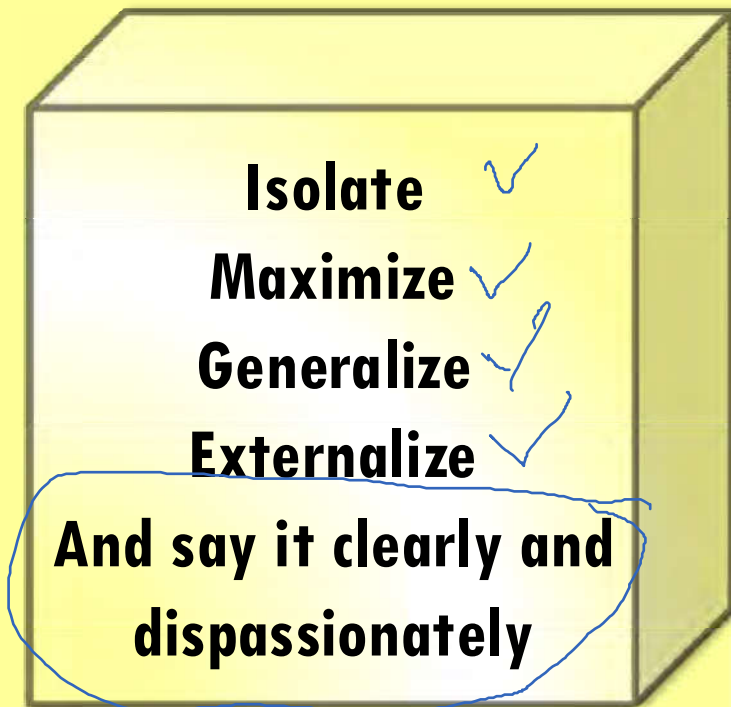
The mission that I prefer is this:



**A bug tracking process
exists for the purpose of
getting the right bugs fixed**

To report a bug well: Isolate


- You have a list of the steps you took to show the error. You're now trying to shorten the list.
- What are the critical conditions?
- Write a report that includes the minimum set of steps needed to replicate the failure
 - Include all the steps needed
- Keep it simple: only one failure per report
- If a sample test file is essential to reproducing a problem, reference it and attach the test file



Isolate the failure:

Eliminate unnecessary steps (1)

- Try taking out individual steps or small groups of steps and see whether you still replicate the failure.
- Sometimes it's not immediately obvious what can be dropped from a long sequence of steps in a bug.

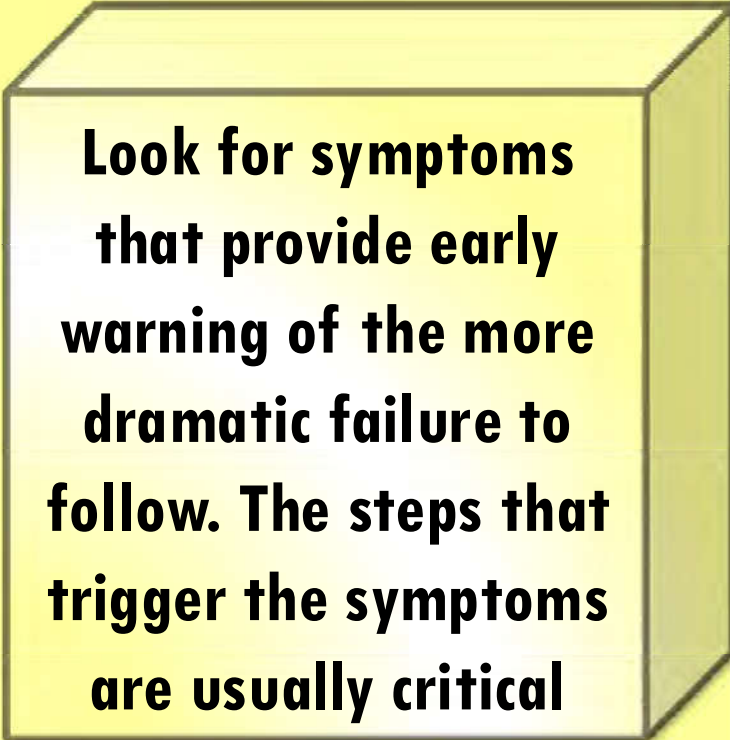
A 3D box with a white front face and a light yellow top and right side. The text is centered on the front face.

**The best report is
the one that reaches
the failure in the
shortest, simplest
set of steps.**

Isolate the failure:

Eliminate unnecessary steps (2)

- Look carefully for any hint of an error as you take each step. Examples:
 - Error messages (you got a message 10 minutes ago. The program didn't fully recover from the error--the problem you see now is caused by that poor recovery.)
 - Display oddities, such as a flash, repainted screen, cursor that jumps back and forth, multiple cursors, misaligned text, slightly distorted graphics, doubled characters, omitted characters, or display droppings (pixels that are still colored even though the graphic that contained them was erased or moved).



Look for symptoms that provide early warning of the more dramatic failure to follow. The steps that trigger the symptoms are usually critical

Isolate the failure:

Eliminate unnecessary steps (3)

- Look carefully for any hint of an error as you take each step:
 - Delays or unexpectedly fast responses
 - Noticeable change in memory used
 - Sometimes the first indicator the system is working differently is that it sounds a little different than normal
 - An in-use light or other indicator that a device is in use goes unexpectedly on or off
 - Debug messages—turn on your system's debug monitor (if it has one)-
-see if/when a message is sent to it

Isolate the failure:

Eliminate unnecessary steps (4)

Once you find what looks like a critical step, try to eliminate almost everything else from the bug report.

- Go directly from that step to the one(s) that appear to be the final trigger(s) for the failure.

If this approach doesn't work, try taking out individual steps or small groups of steps more arbitrarily and see whether you still replicate the failure.

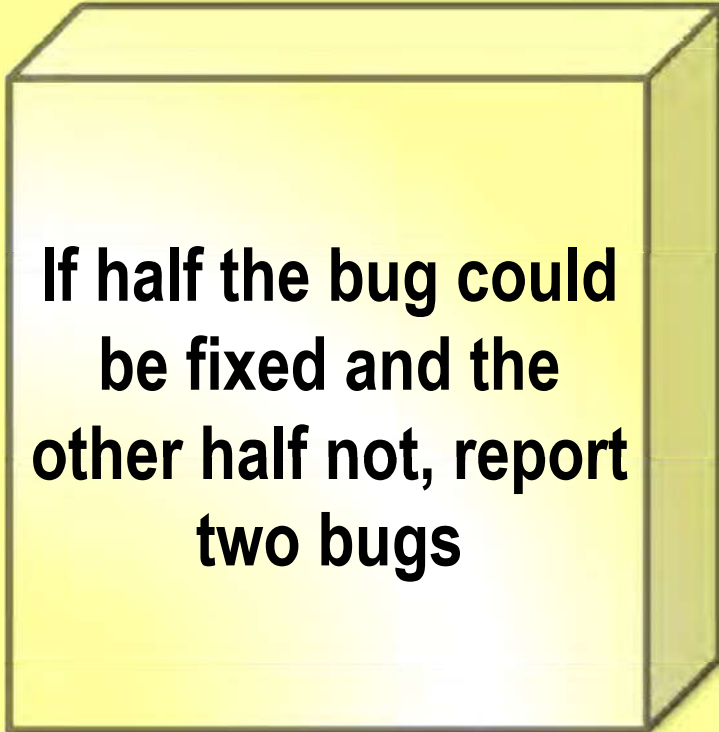
Isolate the failure:

Two failures → two reports.

Reports with two problems:

- Description is longer and harder to follow, and therefore
 - less likely to be addressed and
 - more likely to be misunderstood
- Summary line is often vague (says “fails” instead of describing the failure)
- Half the report gets fixed and it gets closed

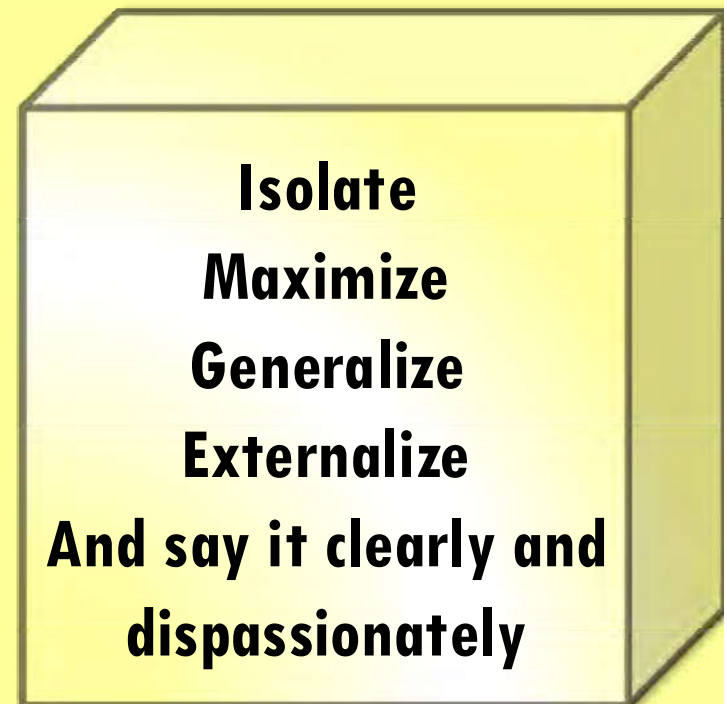
When you report related problems on separate reports, it is a courtesy to cross-reference them.



**If half the bug could
be fixed and the
other half not, report
two bugs**

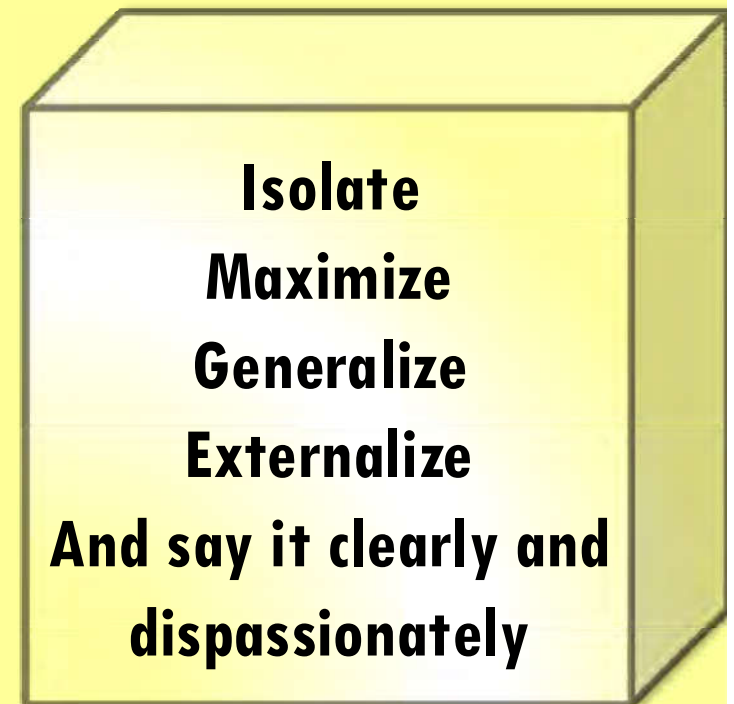
To report a bug well: Maximize

- Follow-up testing to see if you can demonstrate a worse failure than the initial symptom:
 - Vary your actions
 - Vary the program's settings
 - Vary the stored data you give the program
 - Vary the environment



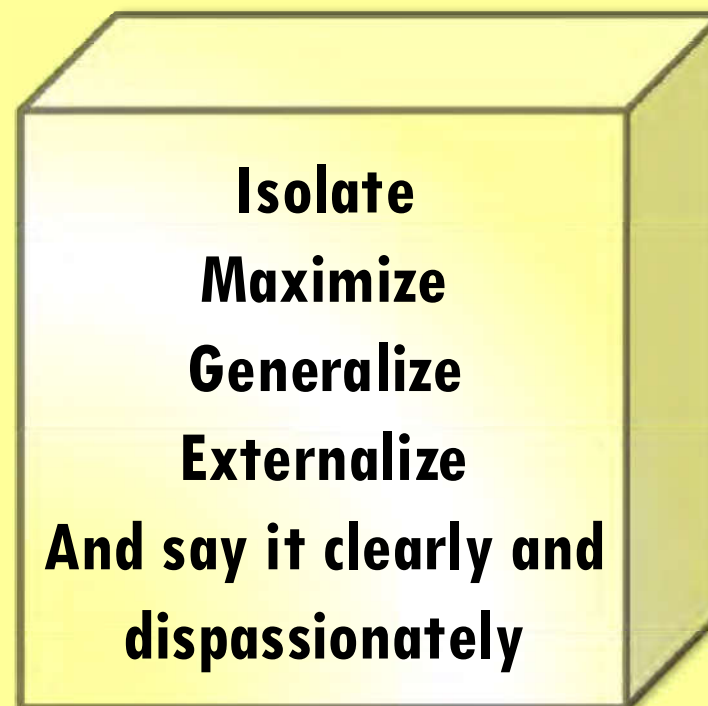
To report a bug well: Generalize

- Uncorner your corner cases (show it fails (or doesn't) under less extreme conditions)
- Show it fails (or doesn't) on a broad range of systems



To report a bug well: Externalize

- Switch focus from the program to the stakeholders
 - What are the consequences of this failure?
 - Is comparative data available?
 - Historical support data for similar bugs?
 - Other historical cost data?
 - Competitors' problems?
 - Have people written about problems like these in this or other products?
 - What benefits does this failure interfere with?
 - WHO WOULD CARE ABOUT THIS FAILURE AND WHY?
 - Get them to help you understand what this costs them.



To report a bug well: Clear and dispassionate

- Make the report easy to understand
- Keep your tone neutral and nonantagonistic
- Angry, blaming reports **discredit** the reporter.

