A developer D creates a piece of software and passes it on for testing and use.

The reporter R uses virtual platform checkpointing to pass the bug to the developer. This ensures perfect replication and that the complete target state is communicated.

The software user finds a bug and needs to report it to the developer. This makes him or her the reporter R.

The developer D creates a piece of software and passes it on for testing and use.
Why? – Bug Reporting is no Fun

When navigating to a page that requires the use of the scroll bars, the scroll wheel on the mouse (and scroll touch bar on the touch pad) fail to scroll the page.

Reproducible: Always

Steps to Reproduce:
1. Navigate to http://www.google.com
2. Enter search query "Waldo"
3. Attempt to scroll using scroll wheel, note failure.

Actual Results:
The window failed to scroll

Expected Results:
The window should move down (if possible) on a scroll down event, or up in a scroll up event.

Currently using the nightly build (Build 20090512041901) with the default theme. Currently using Adblock plus, download statusbar, downthemall!, edit middle, firebug, fireftp, flashgot, google preview, greasemonkey, httpfox, ie tab, microsoft .net framework assistant, nightly tester tools, twitter fox, user agent switcher, and window resizer plugins.

https://bugzilla.mozilla.org/show_bug.cgi?id=492885
Why?

- Guaranteed bug reproduction
  - Any number of times
  - At any point in time, any place in the world, on any host (*)
  - No "cannot reproduce" or "unconfirmed"

- Complete state transfer
  - No need to describe the state
  - No need to reconstruct the state
  - No need to ask follow-up questions
  - No risk that the software used in the bug becomes hard to get

\[=\text{Engineer convenience, process efficiency, faster bug resolution, less development risk}\]
### How? – Technology Background

#### Virtual Platform
- Model of hardware
- Runs the complete SW
- Provides "the impossible"
  - Checkpointing
  - Determinism
  - Synchronous stop
- Speed is of the essence
  - TLM models
  - Fast JIT ISS
  - Minimally sufficient timing detail
- Might be simplified
  - Stubs, partial models, ...
  - As long as the SW runs

#### Checkpoint
- Captures
  - Processor state
  - Memory state
  - Disk state
  - Device state
  - Target time and date
- Does not include
  - Model implementation
  - Session information like breakpoints
- Needs to be portable
  - Across hosts
  - Across implementations
  - Across time and space
Working Efficiently with Checkpoints

- Use differential checkpoints
  - Difference between checkpoint state and initial state
  - Difference between checkpoints
  - Minimize the data moved

- Position the checkpoint time just before bug hits
  - Shorten the wait for developer
  - Minimize the need to provide input scripts and recordings
  - Reporter might have to iterate to find the optimum spot
Working Efficiently: Nightly Boot

Platform team P creates the VP and platform SW setup

The configured and booted target is delivered as checkpoints to all other teams

The reporter gets a second checkpoint from the developer, representing the addition of the developer’s software to the setup

The software configurations and other actions performed by the reporter are captured in a checkpoint and passed back to the developer

Differential checkpoints means the bug state is the addition of all checkpoints passed around
The reporter gets a software package from the developer. This is loaded along with other software on a virtual platform that the reporter has also (re)configured.

The bug report checkpoint contains both hardware and software changes from the platform team baseline.

Note the order of operations. This puts the developer’s software in a new hardware setup.
Replaying Target Stimuli

Boot...

Configure...

Run tests...

Bug!

Note that many different tests can be started from this checkpoint.

Inputs occurring after the last checkpoint was taken, but before the bug hits.

Recording of last few inputs.

Merged checkpoint and the recording is the bug report contents.

Checkpoint merge.
Quick Demo Example
Paper Addendum Online

- http://jakob.engbloms.se/archives/1231
  - More related work
WIND RIVER