# CSS Deployment @ NSLS2
## Lesson Learned

Michael Davidsaver
Presented by: Gabriele Carcassi

# Overview

- Based on experiences using and supporting CSS @ NSLS2 beginning in Fall 2011.

- Deployment process

- Operational issues

  - Software

  - People

# The Problem

- 2 Control Rooms (soon to be 3)
  - Between 2-5 computers in each.
- 3 multi-user remote access systems
  - NoMachine NX
- 37 concurrent users (26 Apr 2013)
  - 23 CSS instances running
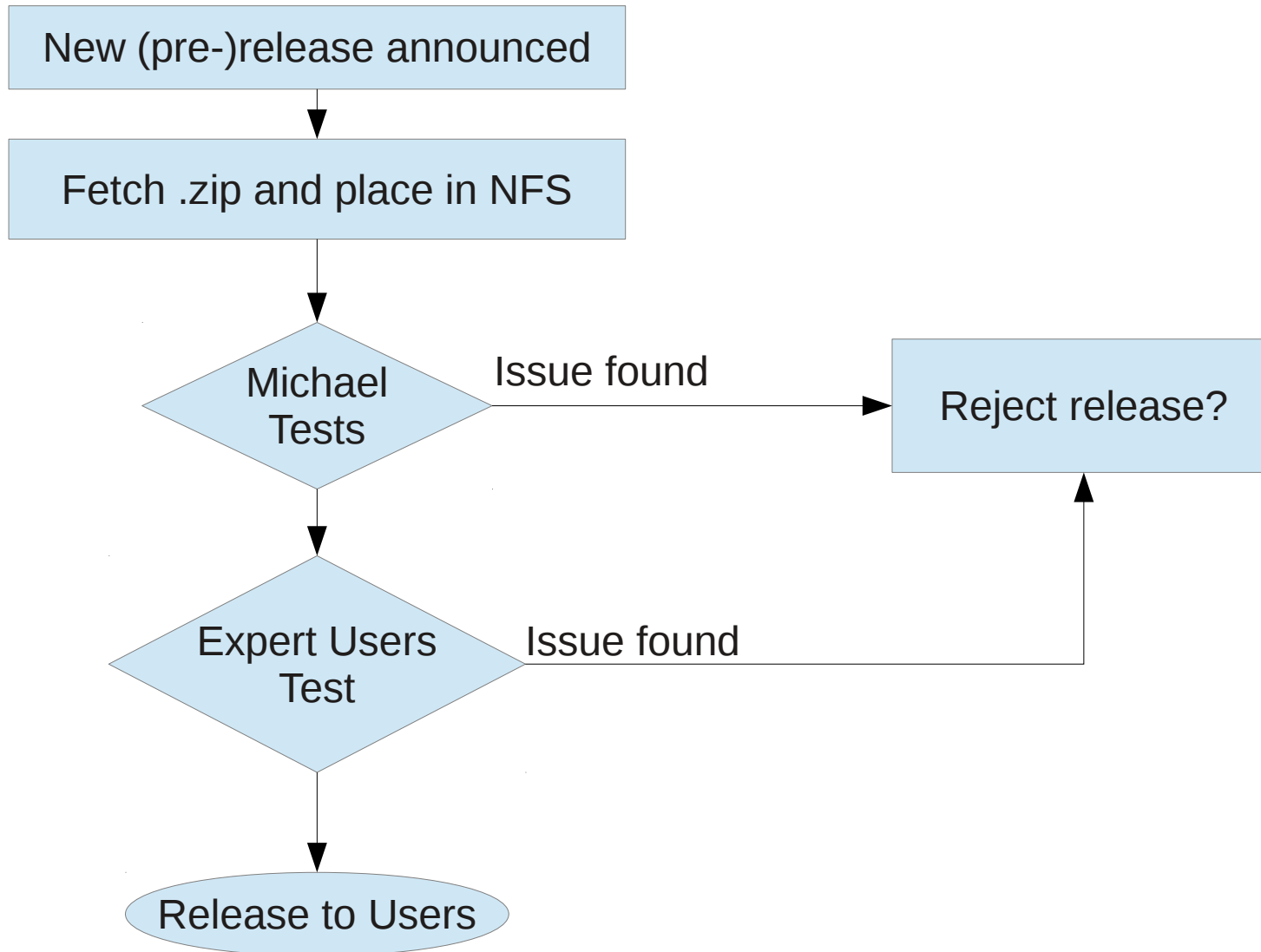  - (A slow day)

# Deployment goals

- Minimize the number of versions in use.

  - Fewer configurations to test

- Ensure that non-experts use tested version.

- Allow quick fallback to a previous version

- No first-time setup.

  - Quick and friendly to non-experts

# Technique

- Files served through NFS
  - All read-only!
  - Binaries and .opi files tracked separately
- Wrapper script to execute
  - Provide machine/version specific preferences file
  - Extra lock file and log file to find "lost" instances
  - "run-css"
- Several CSS-NSLS2 versions are available.

# Deployment Process

New (pre-)release announced

Fetch .zip and place in NFS

Michael Tests → Issue found → Reject release?

Expert Users Test → Issue found → Reject release?

Release to Users

# BOY Deployment in practice

- Several versions can be in use at the same time

  – With the same .opi files. Compatibility is critical!

- Operations uses "stable" (older version)

- Controls developers using newer version

- Typically 1 month to evaluate a new release

  – Manual GUI testing is slow

# Michael's test procedure

- Targeted to BOY (most important tool for us)
- Side-by-side comparison of commonly used screens in both versions.
  - Eyeball diff is very selective
  - Poor coverage
  - Runtime: ~15 min
- Use the new version for a while
  - Includes expert/detail screens
  - Limited to the screens Michael uses
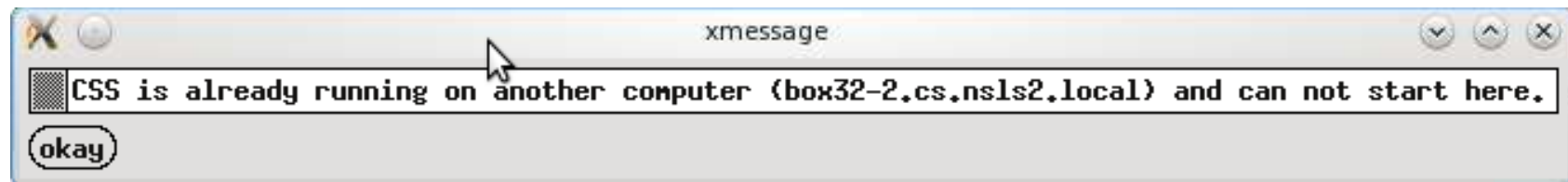  - Runtime: ~1 month

# Past regressions found

- BOY

  - Change to property type causes all existing boolean buttons to request confirmation.  Not critical, fix with script. 3.1.0 in production.

  - Linking container draws differently when .opi read from disk vs. HTTP.  Critical. 3.1.1 skipped.

  - Change to plots with log scale gives labels in fixed precision decimal (vacuum always 0.000000).  Not critical, fix with script. 3.1.2 in production

  - 3.1.3 and 3.1.4 not considered due to time constraints

  - Priority of top level (Display) macros changed.  Critical. Skipped 3.1.5

# Operational issues

- Eclipse workspace + NFS home directory = "workspace already in use"

  - Users frequently leave CSS running on one console, then try to start it again from another.

  - Our **#1** support issue!

  - "Solved" with wrapper script.

  - Not perfect.  Wrapper script can't stop running instance, or identify "stale" lockfiles.

xmessage

CSS is already running on another computer (box32-2.cs.nsls2.local) and can not start here.

okay

# Supporting CSS

- Most issues encountered in the control room.
  - RTFM is not well received at this point
- Support by phone is first line.
  - "I need to see what the user sees!"
  - Stateful GUI complicates

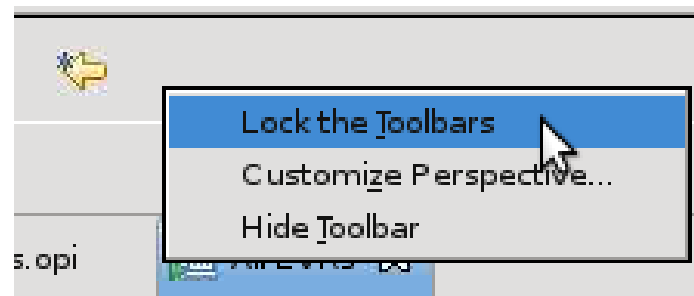# Suggestions for Improvement (1)

- Menu item and/or hot key for "Open Top OPI in run mode".
  - The single most important button in CSS.
  - Difficult to describe over the phone.
    - Our **#2** support issue
  - Has moved between releases
  - Toolbar unintentionally/moved closed by users
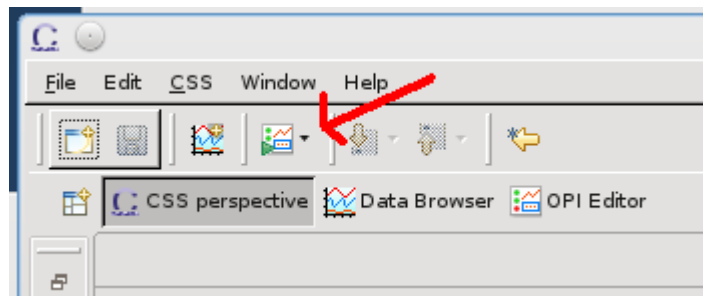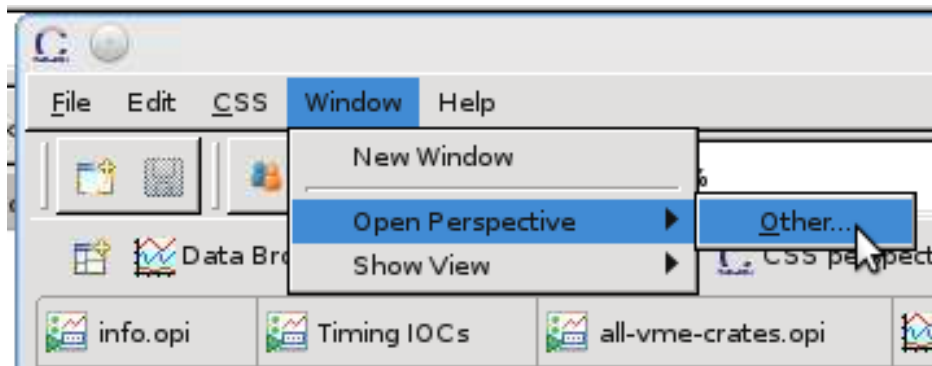


As seen on postings in NSLS2 control rooms

# Suggestions for Improvement (2)

- Lock toolbars by default
  - Non-expert users unintentionally move or close
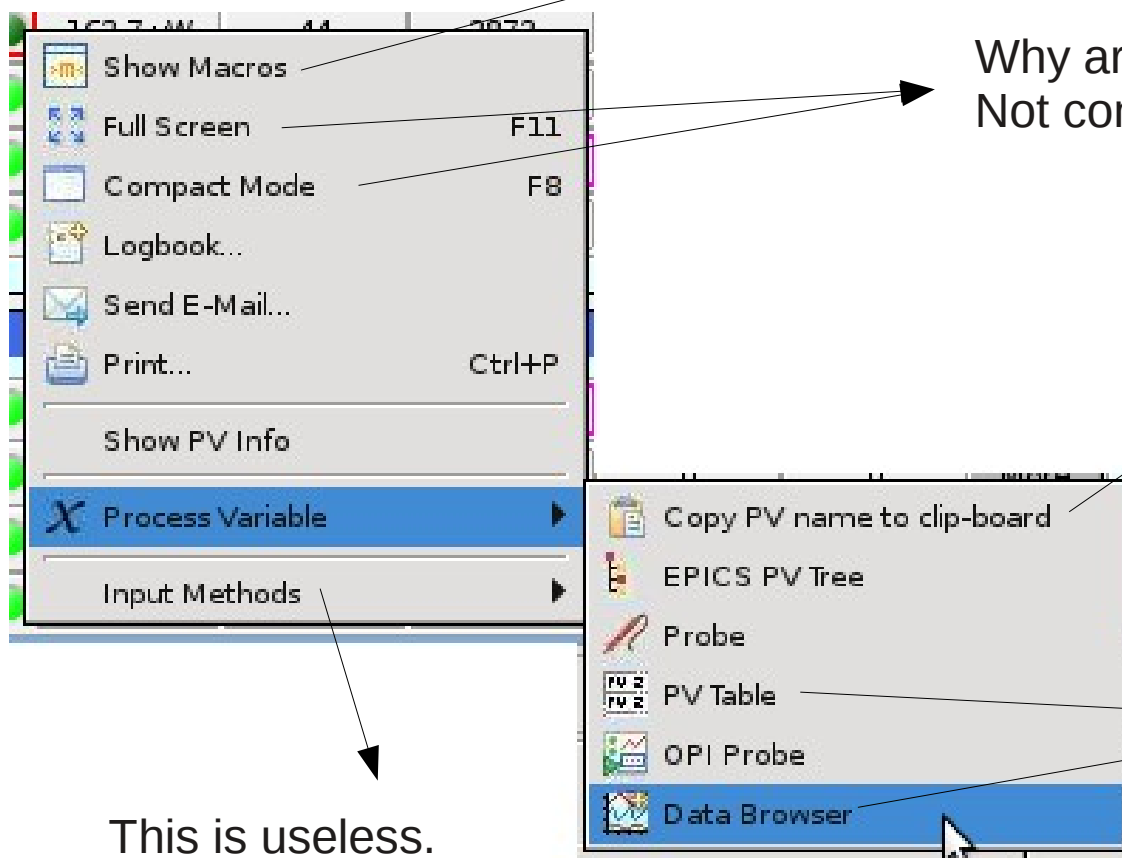
# Suggestions for Improvement (3)

- Reduce number of mouse clicks
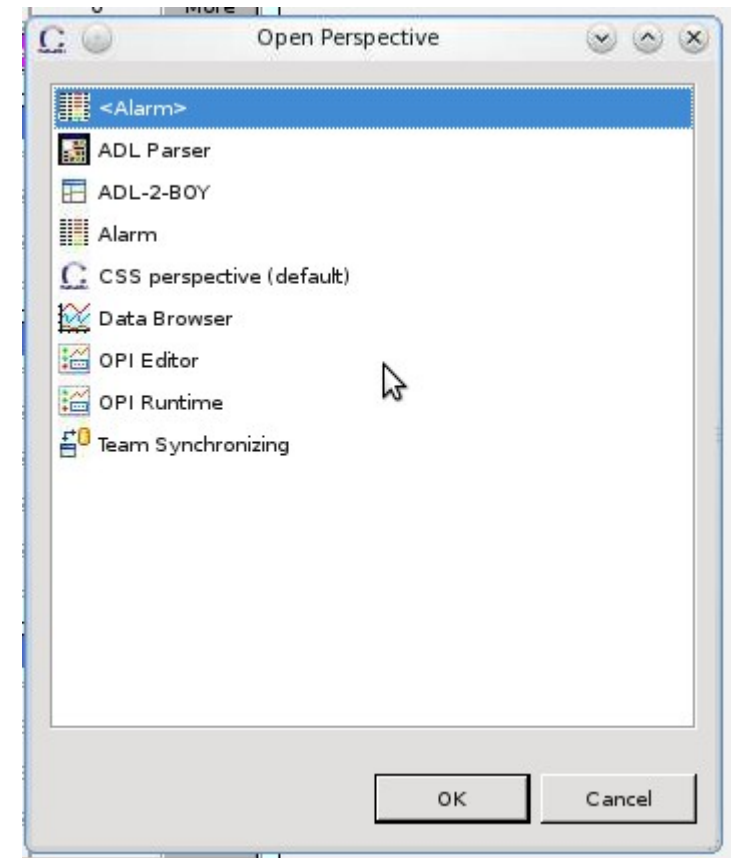


**Can the list of perspectives appear directly in a menu?**

# Suggestions for Improvement (4)

- Improve context menu

Useful for debugging, but not for users.
Move this to bottom.

Why are these in the context menu?
Not contextual...

**The most useful entry!
Move to top of parent
menu for faster access?**

Always starts a new table/plot.

How to add PV to existing
table/plot w/o copy+paste?

This is useless.
Can it be hidden?

# Conclusion?

- Difficulties in deploying CSS
  - Testing
    - BOY is our most important control room tool
    - Heavy usage means small issues have big impacts
  - Support
    - Eclipse based UIs have significant learning curve