Deploying CS-Studio at DLS

Will Rogers, High Level Applications
Topics

- Why CS-Studio?
- Work done
- Decisions
- Work to do
Why CS-Studio?

- DLS has many EDM screens
- EDM is approaching end of life
- DLS has two other Eclipse RCP projects
Why CS-Studio?

• Adopting CS-Studio would allow:
  – integration of different products
  – sharing of expertise
  – sharing of infrastructure

• BUT converting from EDM to CS-Studio is not trivial...

• We need to transfer with minimal disruption
Work done

- EDM-style windows in CS-Studio
- ...coming soon?
Work done

- EDM conversion accuracy
Work done

- Integration of EPICS archiver and custom fast archiver
Work done

- New-style beamline controls screens
Work done

- IFM (non-EPICS engineers) love CSS
Work done

• Converting Python-Qt screens to CSS
Decisions

• We need to integrate with the existing DLS release process
• We need a manageable transition (?!?)
• How to:
  – lay out OPI files?
  – handle relative links?
  – launch OPI screens?
  – execute the final conversion?
Decisions

- Launch EDM-style screens using existing launcher
Decisions

• Allow BOY and EDM to co-exist
• Convert .opi files alongside .edl files in existing filesystem structure
• Allow developers to choose when to switch from EDM to BOY
• Commit to VCS and edit BOY screens in future
Work to do

- Repeatable local CS-Studio build
- Publish roll-out plan
- Chase down remaining niggles
- Release versions 4.1 and 4.2 at DLS
- User and developer training
Thanks!

• Skepticism and questions very welcome...

@wrgrs