

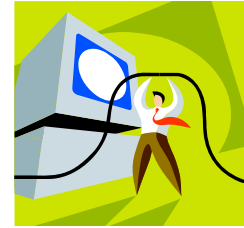
Common Software Tools

SKA

Common Software Process

Christopher.williams@oerc.ox.ac.uk

The Case For A Common Software Engineering Infrastructure



- Large scale software development is difficult and projects regularly fail.
- Projects can benefit from expertise in non-core areas e.g.
 - Packaging and distribution. System Administration
 - Configuration Management, Release management
 - Quality Assurance, Technical Documentation, Tool support
 - Development Environments, Dependencies, Design, Reviews



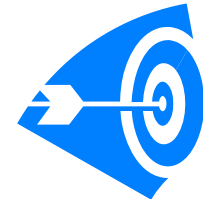
- Cost effective
 - avoid duplication of effort
 - share resources

#1 Reason – Happy Clients



- Dedicated resources can do a better job
- Timely support
- Quality Control – more likely to work
- Enabling technology – developers can achieve much more

Software Processes



- **The function of process is to trap problems**
 - Sooner problems are unearthed the easier and cheaper they are to fix.
- Process does not have to be tedious
 - Tool support
 - Increased productivity
- Process needs buy in to be effective
- Process needs regular review to ensure it is meeting its aims
 - Bad processes are counter-productive
- Suggest a “Process Working Group”

Common Software Tools



- Packaging and distribution
 - e.g. The Oxford Repositories
 - Platform provisioning service, overnight build service
 - Issue Tracking and Configuration management tools
 - Common Build Environment
 - Quality Assurance Tools
 - Requirements Tracking Tools (IBM Doors)
-
- Integration of these tools to optimise efficiency and support our process
 - Can we define a Common Architecture?
 - Platforms, dependencies, programming languages

