

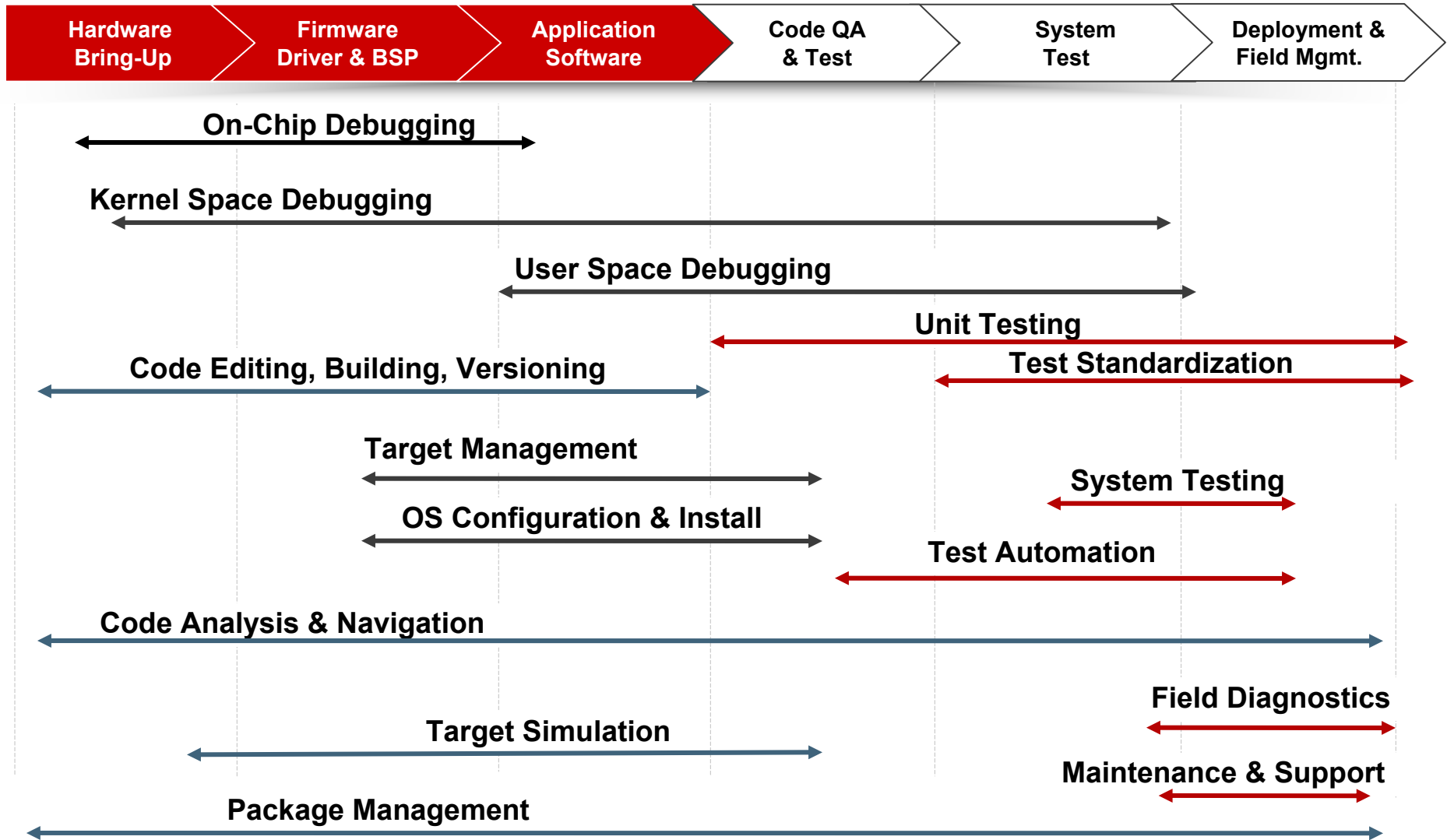
The image shows a close-up, low-angle shot of the Wind River logo on a light-colored building facade. The logo consists of the words "WIND RIVER" in large, bold, red, three-dimensional block letters. The letters are mounted on the wall and cast shadows. The sky above is a clear, bright blue. The perspective is from below, looking up at the building.

WIND RIVER

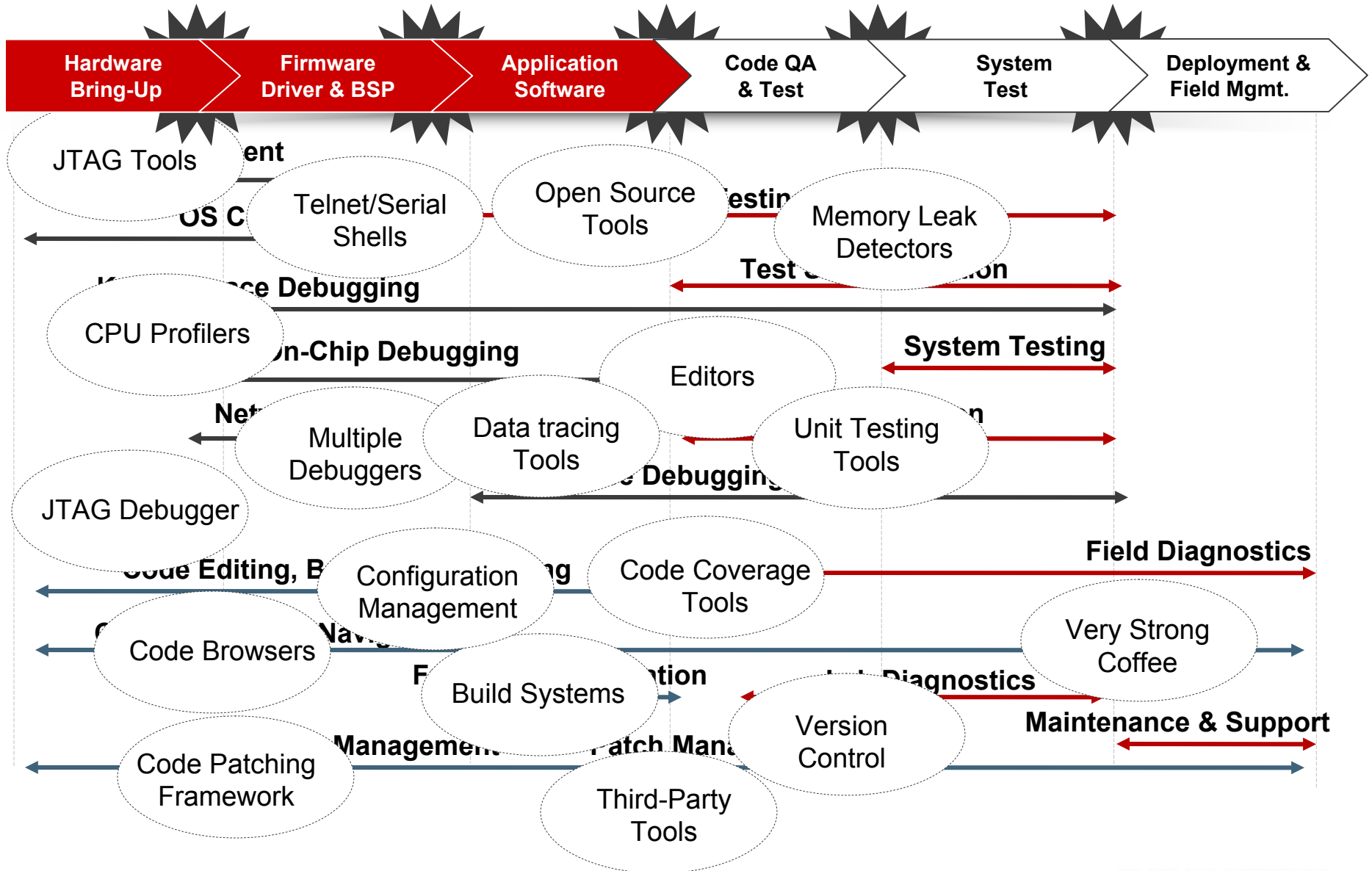
How Good are Open Source Development Tools

Glenn Seiler
Sr. Director, Market Development

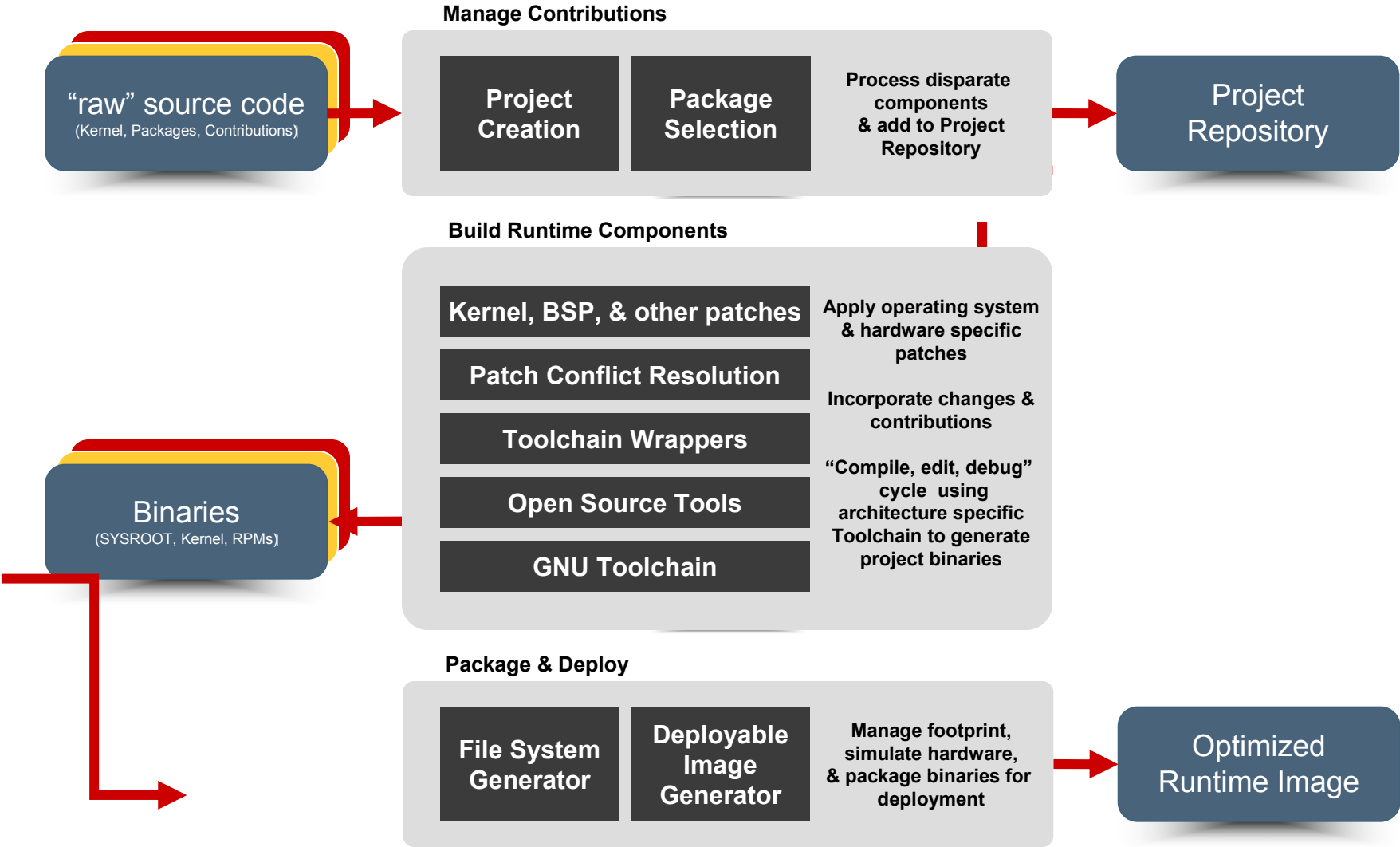
The Development Life Cycle



Diversity of Tools



Typical Build Cycle



**Today's tools are
getting the job done**

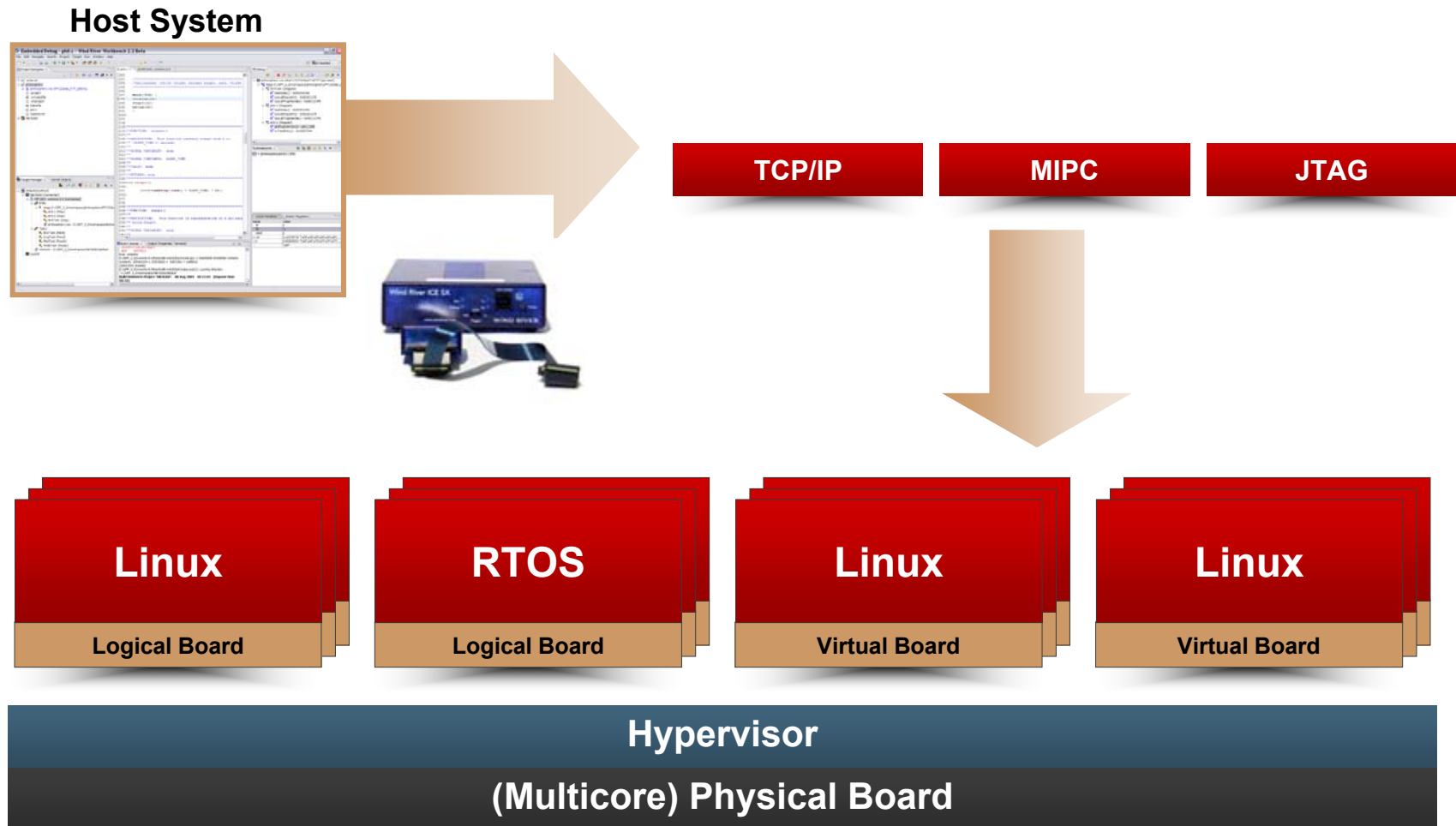
BUT.....

Multicore changes everything

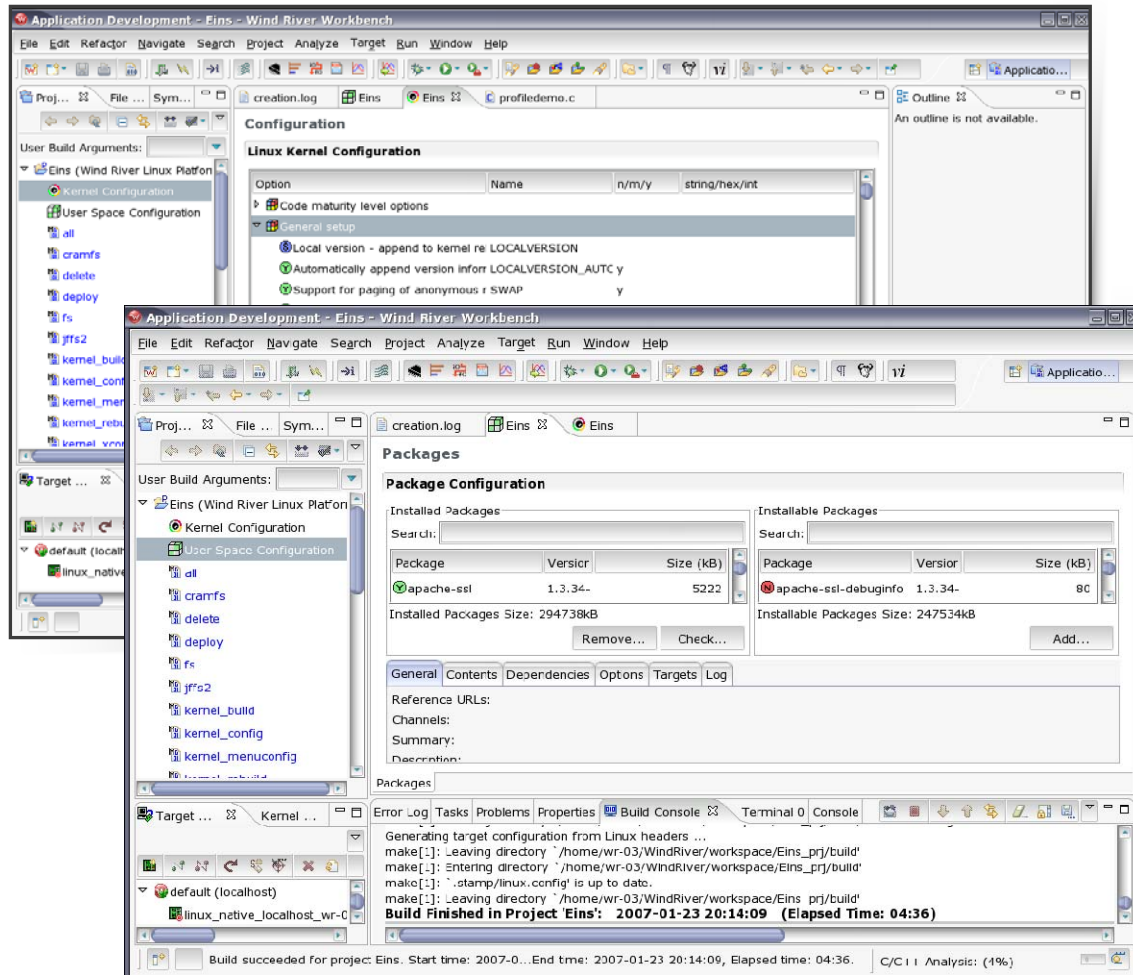
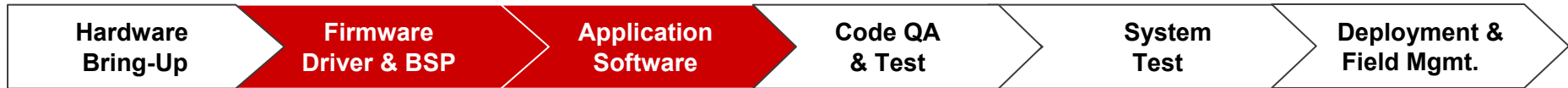
Challenges of Debugging with Multicore

- **How to effectively manage shared resources, such as memory and peripherals**
- **How to debug OS and application code over multiple cores, boards, and systems**
- **How to synchronize on-chip debugging across multiple cores served by one or more JTAG connections**
- **How to debug homogeneous and heterogeneous cores on a single die, then coordinate the debugging over an entire system**
- **How to use JTAG debugging effectively with agent-based debug and ensure a smooth handoff between different debug tasks**
- **How to optimize use of the JTAG interface for responsiveness and throughput**

Even harder with Virtualization and Multicore

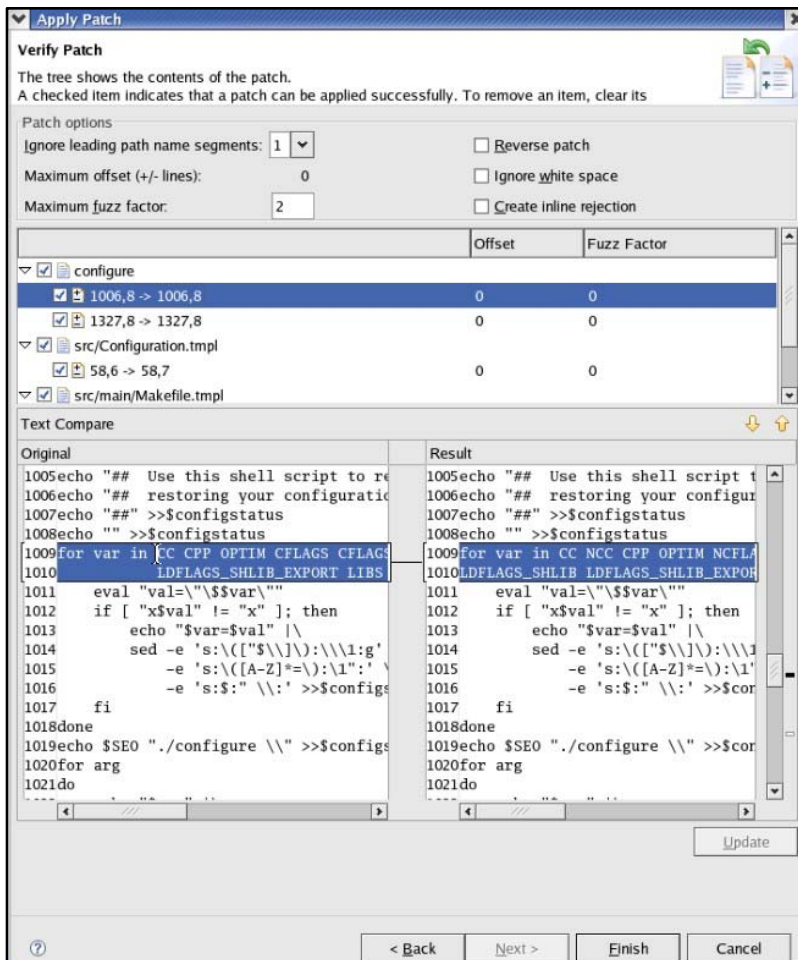


Kernel & User Space Management



- **Linux Kernel Configuration Editor**
 - View and modify the kernel configuration
 - Include and exclude components
 - Find and view dependencies
- **Linux User Space Configuration Editor**
 - View, add, and remove RPMs from the target file system
 - Build packages
 - Prepare packages for patching

Patch Management



Linux Patch Model

- Never touch the original source
- Provide changes as patches
- Patches allow you to track changes and versions

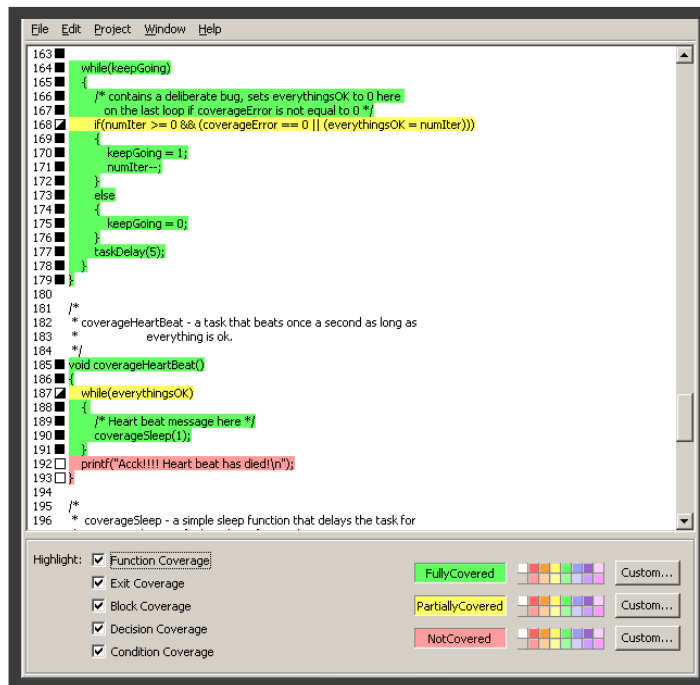
Patch management:

- Classic open-source tools (`diff`, `patch`, and `quilt`)
- Browse, apply, and preview patches
- Vendor specific enhancements such as highlighting patched code and displaying the patch number

Analyze & Visualize Behavior



A set of **dynamic visualization and analysis tools** to help uncover memory leaks, do performance analysis, track changes in system variables, and locate untested code



CoverageScope is now available for Linux

- Helps complete your test suite by showing what's been tested and what hasn't
- Important tool for improving your product quality—now available for Workbench 2.6

Summary - The Open Source Tools Cycle

