

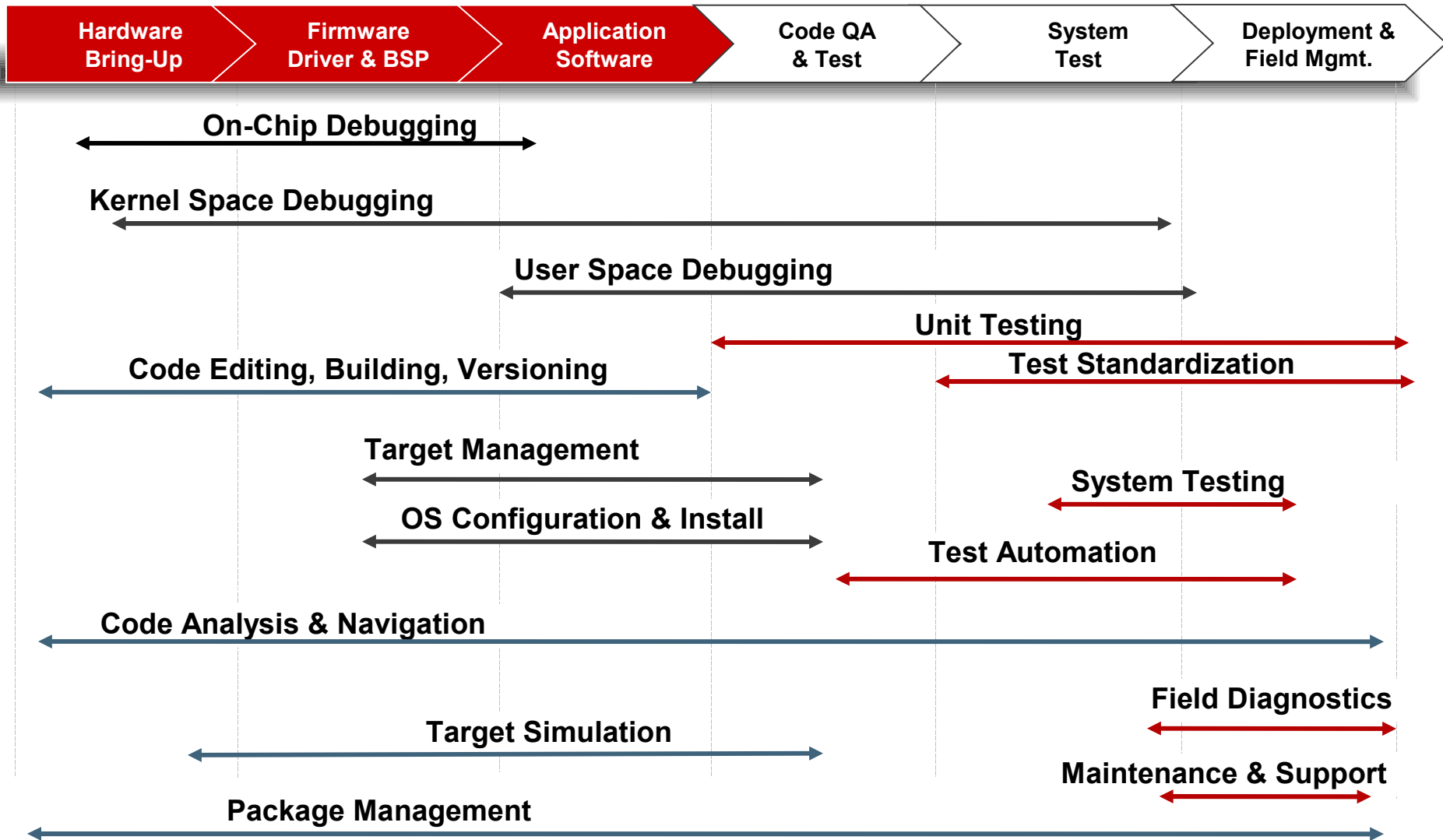
The image shows the exterior of a building with the words "WIND RIVER" mounted in large, red, three-dimensional block letters. The letters are set against a light-colored, possibly concrete or stone, wall. The sky above is a clear, bright blue. The perspective is from a low angle, looking up at the building.

WIND RIVER

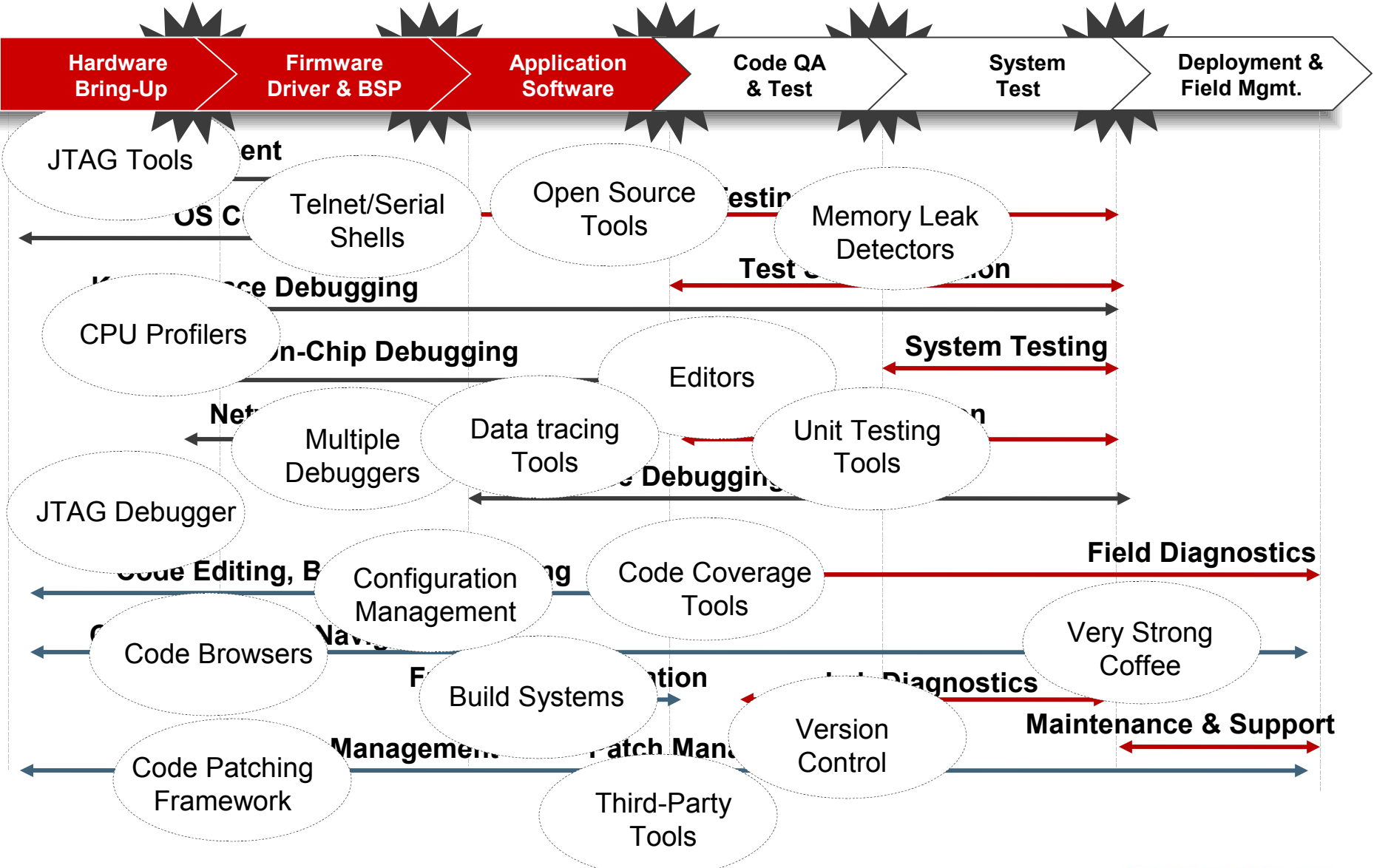
How Good are Open Source Development Tools

Glenn Seiler
Sr. Director, Market Development

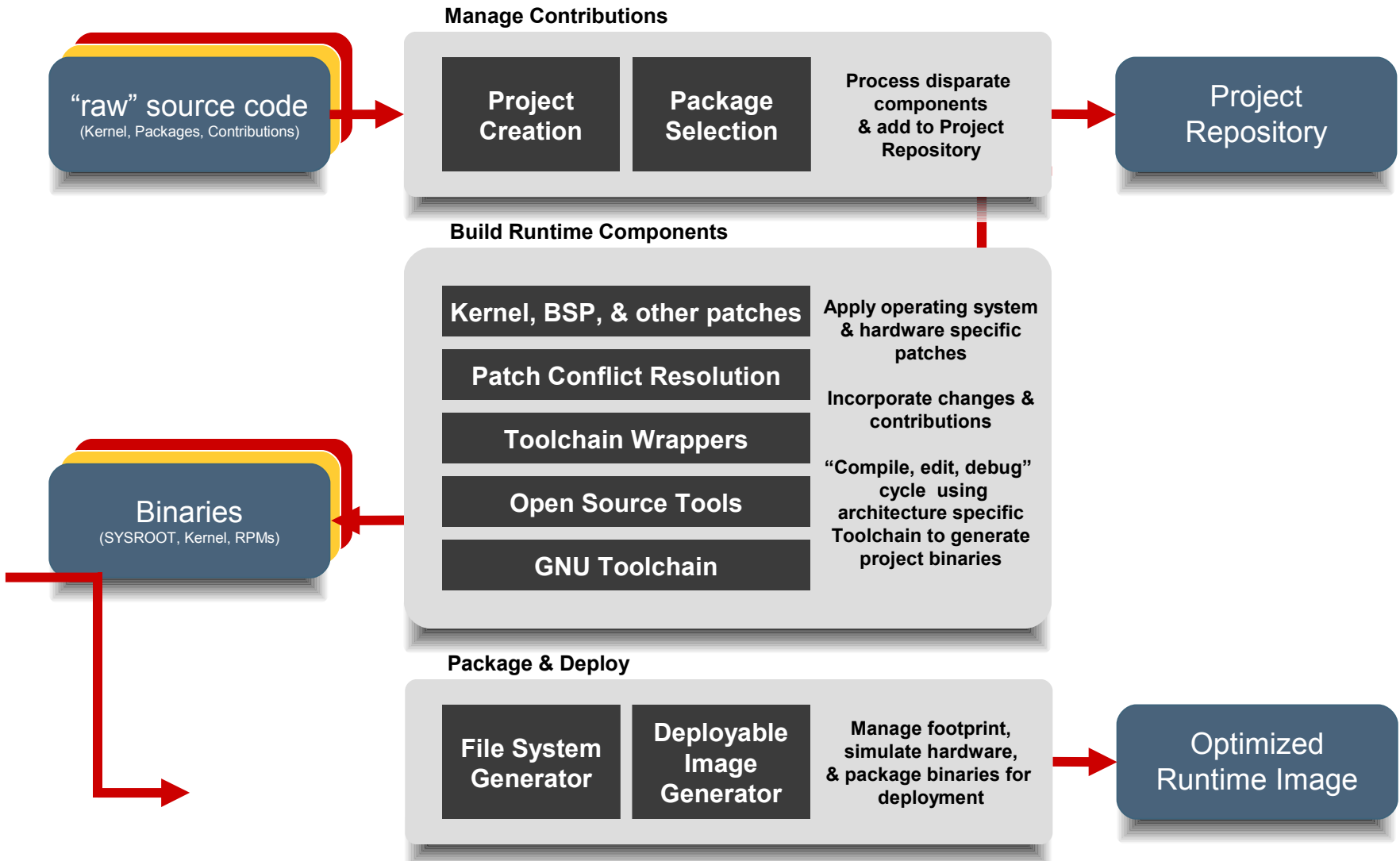
The Development Life Cycle



Diversity of Tools



Typical Build Cycle



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

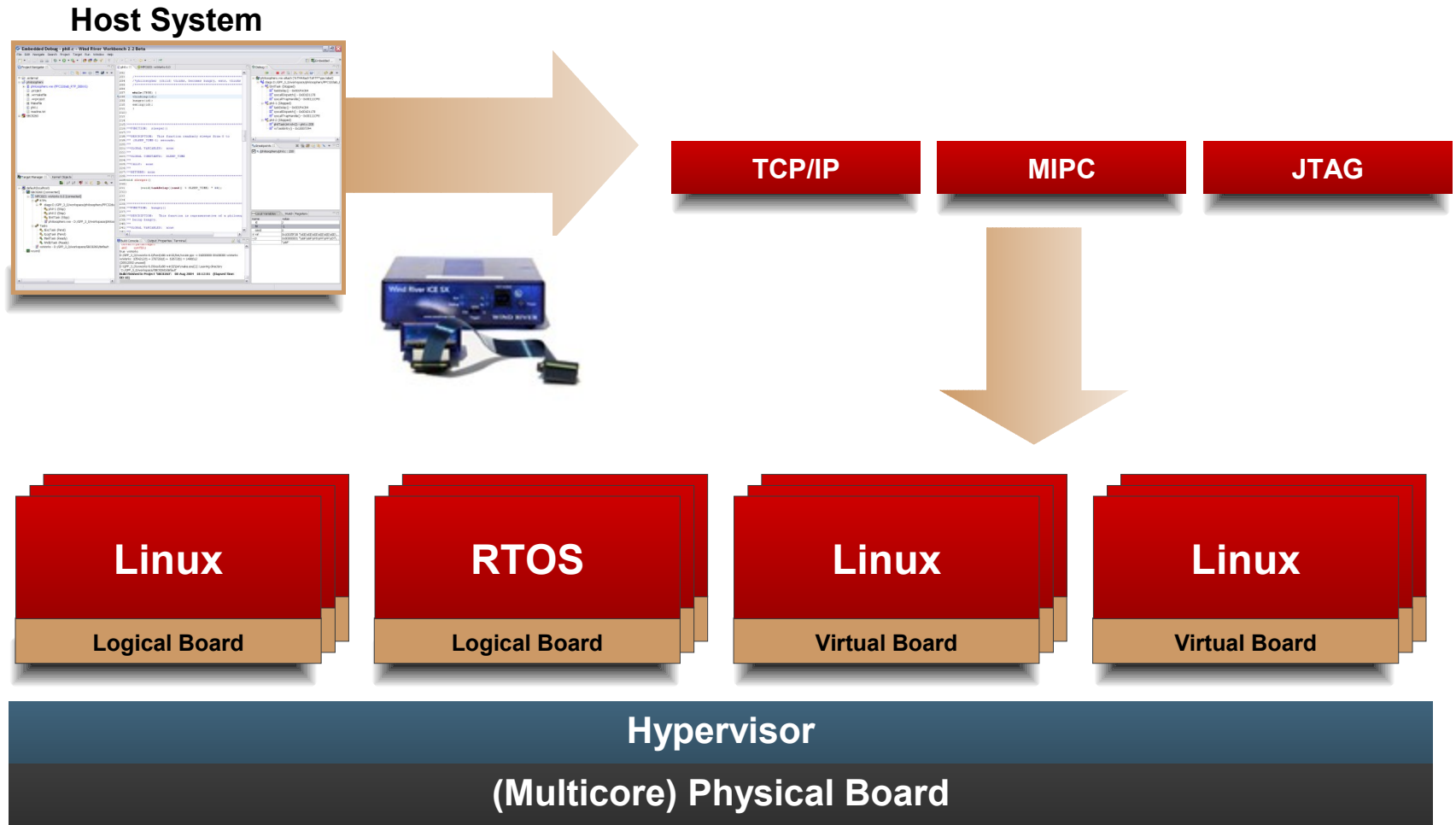
BUT.....

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

Hardware
Bring-Up

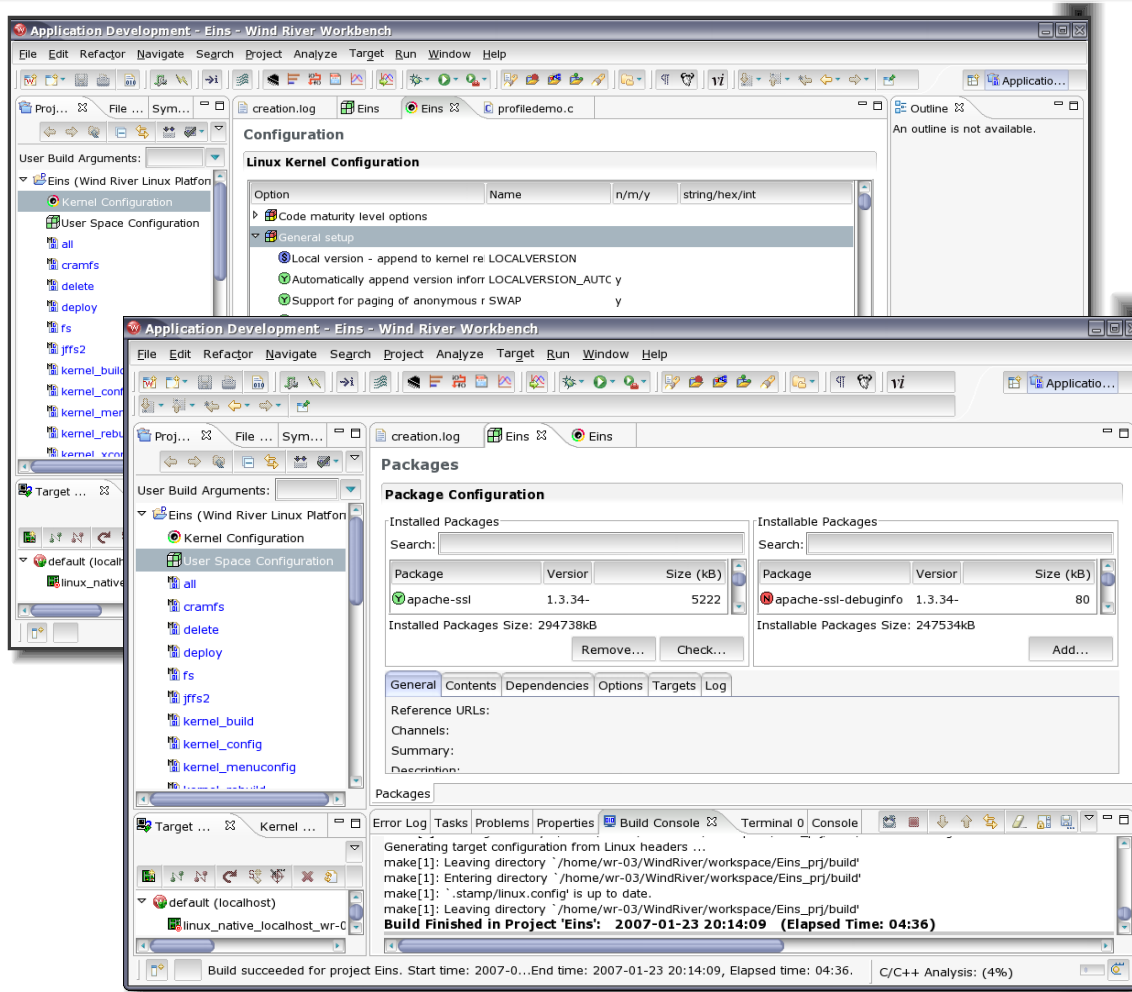
Firmware
Driver & BSP

Application
Software

Code QA
& Test

System
Test

Deployment &
Field Mgmt.



- **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

Hardware
Bring-Up

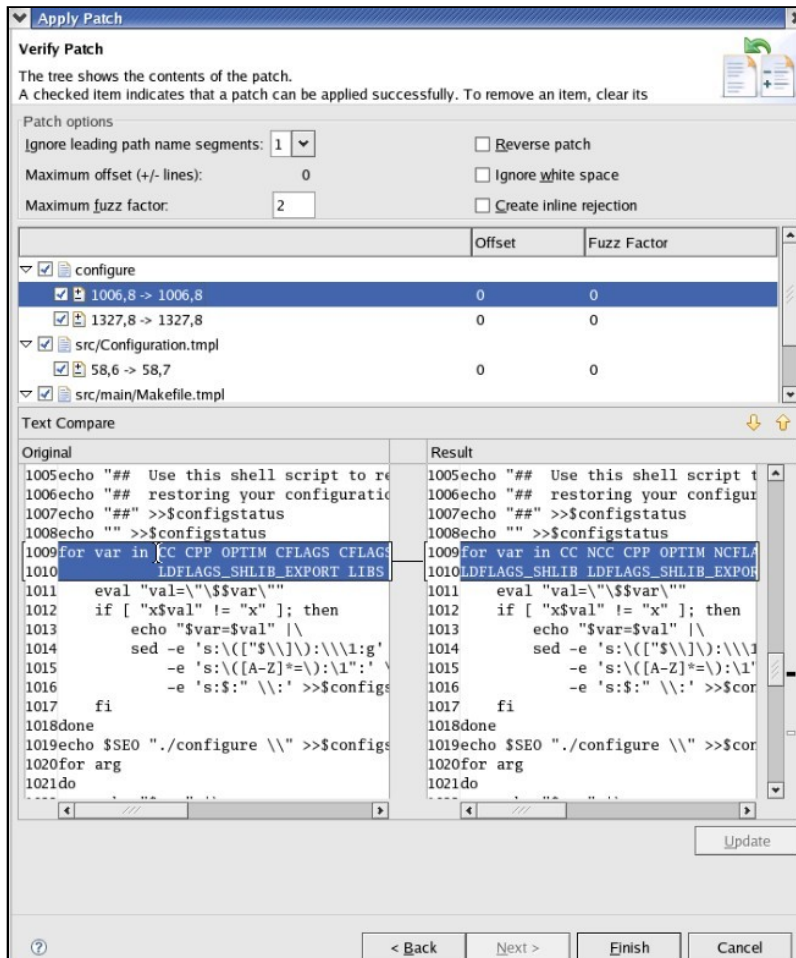
Firmware
Driver & BSP

Application
Software

Code QA
& Test

System
Test

Deployment &
Field Mgmt.



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

Hardware
Bring-Up

Firmware
Driver & BSP

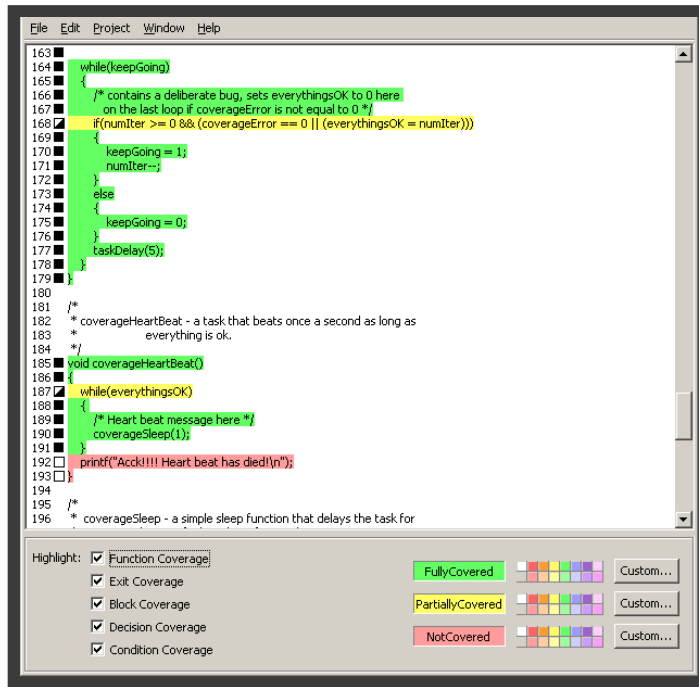
Application
Software

Code QA
& Test

System
Test

Deployment &
Field Mgmt.

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



```
163 while(keepGoing)
164 {
165     /* contains a deliberate bug, sets everythingOK to 0 here
166     on the last loop if coverageError is not equal to 0 */
167     if( (numiter >= 0 && (coverageError == 0) || (everythingOK = numiter))
168     {
169         keepGoing = 1;
170         numiter--;
171     }
172     else
173     {
174         keepGoing = 0;
175     }
176     taskDelay(5);
177 }
178 }
179 }
180
181 /*
182  * coverageHeartBeat - a task that beats once a second as long as
183  * everything is ok.
184  */
185 void coverageHeartBeat()
186 {
187     while(everythingOK)
188     {
189         /* Heart beat message here */
190         coverageSleep(1);
191     }
192     printf("Ack!!! Heart beat has died!\n");
193 }
194
195 /*
196  * coverageSleep - a simple sleep function that delays the task for
```

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

