

# National Ignition Facility

## Software Engineering Processes Used to Develop the NIF Integrated Computer Control System\*

*A. Peter Ludwigsen*



2007 International Conference on Accelerators and Large Experimental Physics Control Systems, Knoxville, TN,  
October 14, 2007

Lawrence Livermore National Laboratory, USA

\* This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

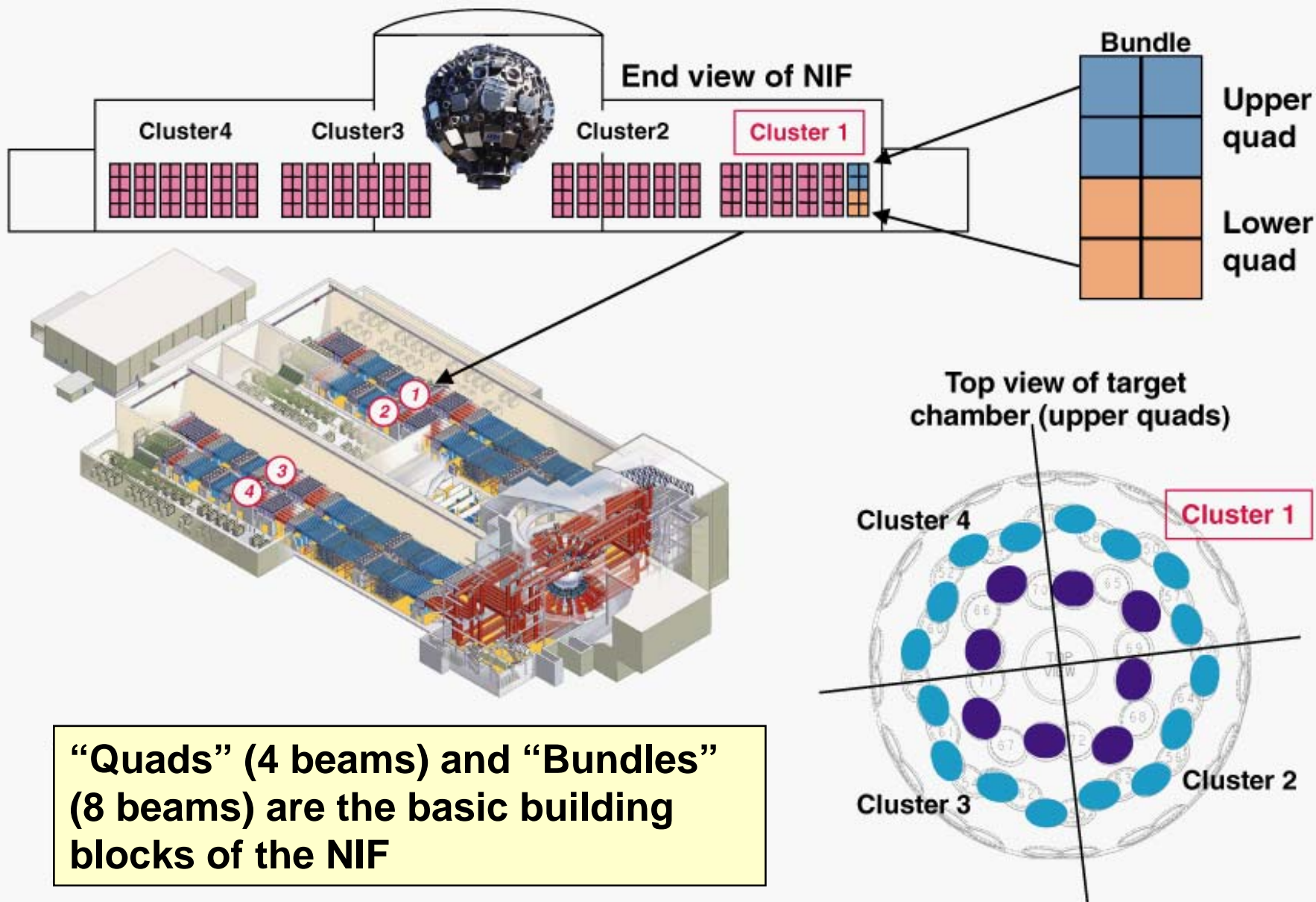


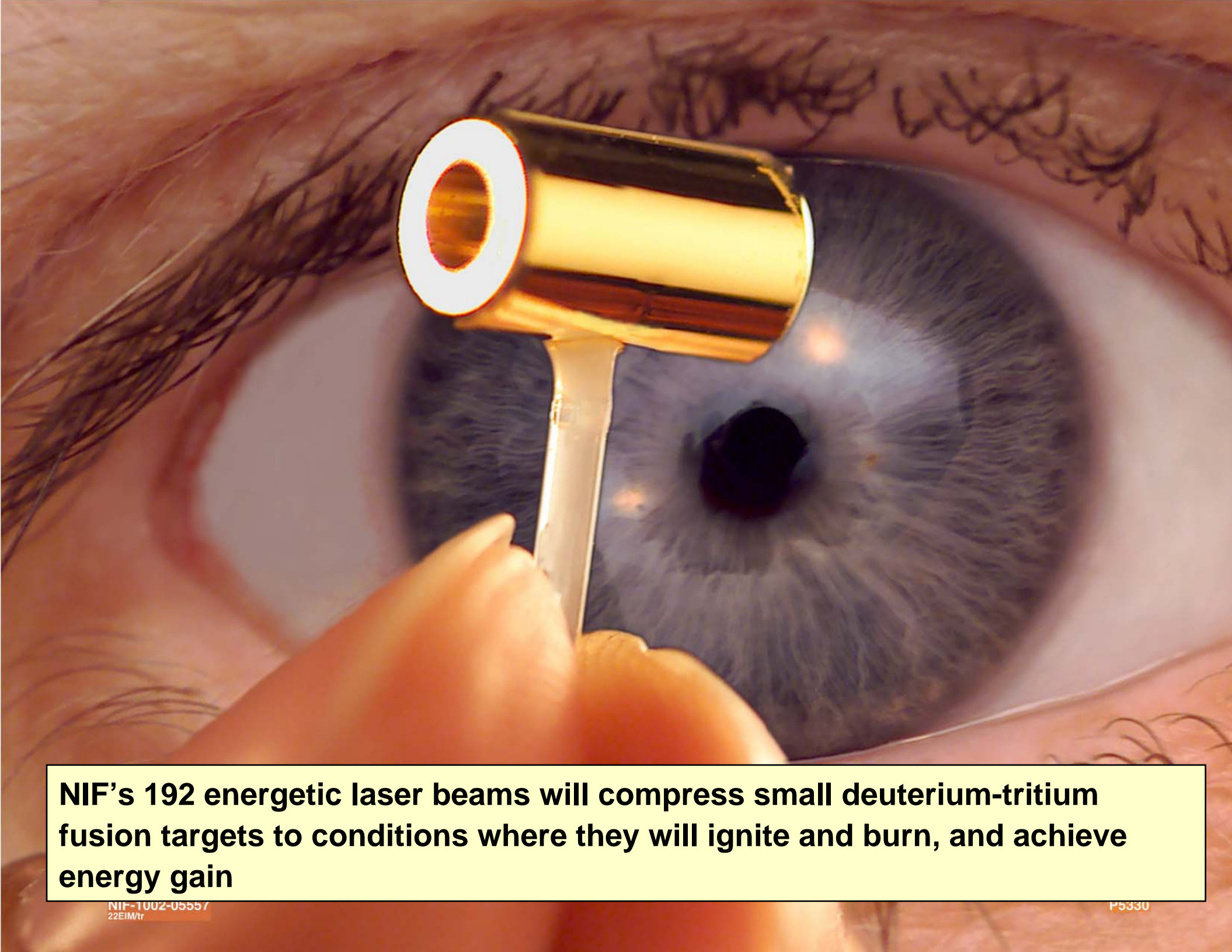


- NIF is a stadium-sized facility that will contain**
- **192-beam, 1.8-Megajoule, 500-Terawatt, ultraviolet laser system**
  - **10-meter diameter target chamber with room for nearly 100 diagnostics**



# NIF is a 192 beam laser organized into “clusters”, “bundles”, and “quads”





**NIF's 192 energetic laser beams will compress small deuterium-tritium fusion targets to conditions where they will ignite and burn, and achieve energy gain**



# The Integrated Computer Control System (ICCS) robustly operates a complex physics facility



The National Ignition Facility

- **Control 60,000 points**
  - 1,500 processes
  - 850 computers
  - 140,000 distributed objects
- **Align and fire 192 lasers**
  - < 50 microns on target
  - < 30 psec timing
- **Automatic shot controls**
  - < 4-hours shot-to-shot
  - Hands-off operation
- **Assure**
  - Situational awareness
  - Machine protection

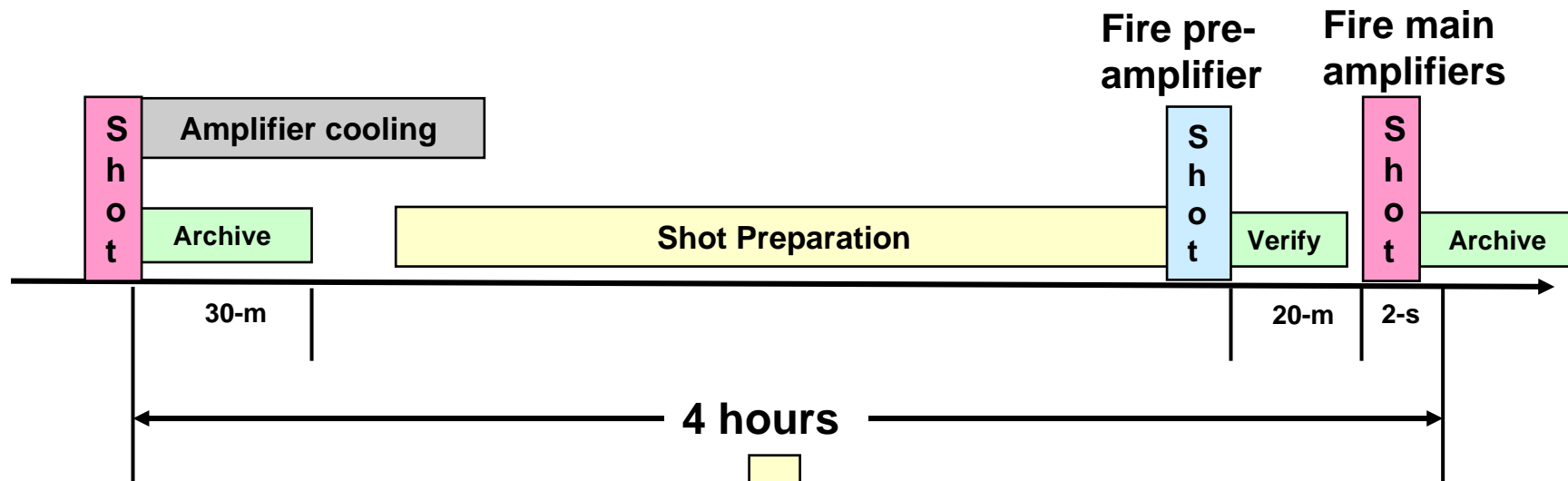
**Automatic control of 192-beam shots are overseen by 14 operator stations**



# ICCS is required to automatically align, fire and diagnose laser shots every 4 hours



The National Ignition Facility



## Automated Shot Cycle

- Acquire campaign shot goals from laser physics model
- Perform automatic alignment and wavefront correction
- Configure diagnostics and laser performance settings
- Conduct countdown (software:4-min and timing:2-sec)
- Assess shot outcome and archive shot data

# Summary of Key Technical Challenges



The National Ignition Facility

Key Challenges	Solutions
Scale from single bundle to full NIF	Partitioned into 24 independent bundles
Align 192 beams on target in 30 minutes	Robust parallel image processing of machine vision sensors
Fire laser shots in 4-hour cycles	Shots automated with model-based control software
Assure machine protection	Incorporated shot verification and situational awareness features
Provide flexible controls capable of supporting NIF for decades	Created versatile plug-in software framework using CORBA distribution

# Software engineering and computer science were key to building NIF's large-scale control system



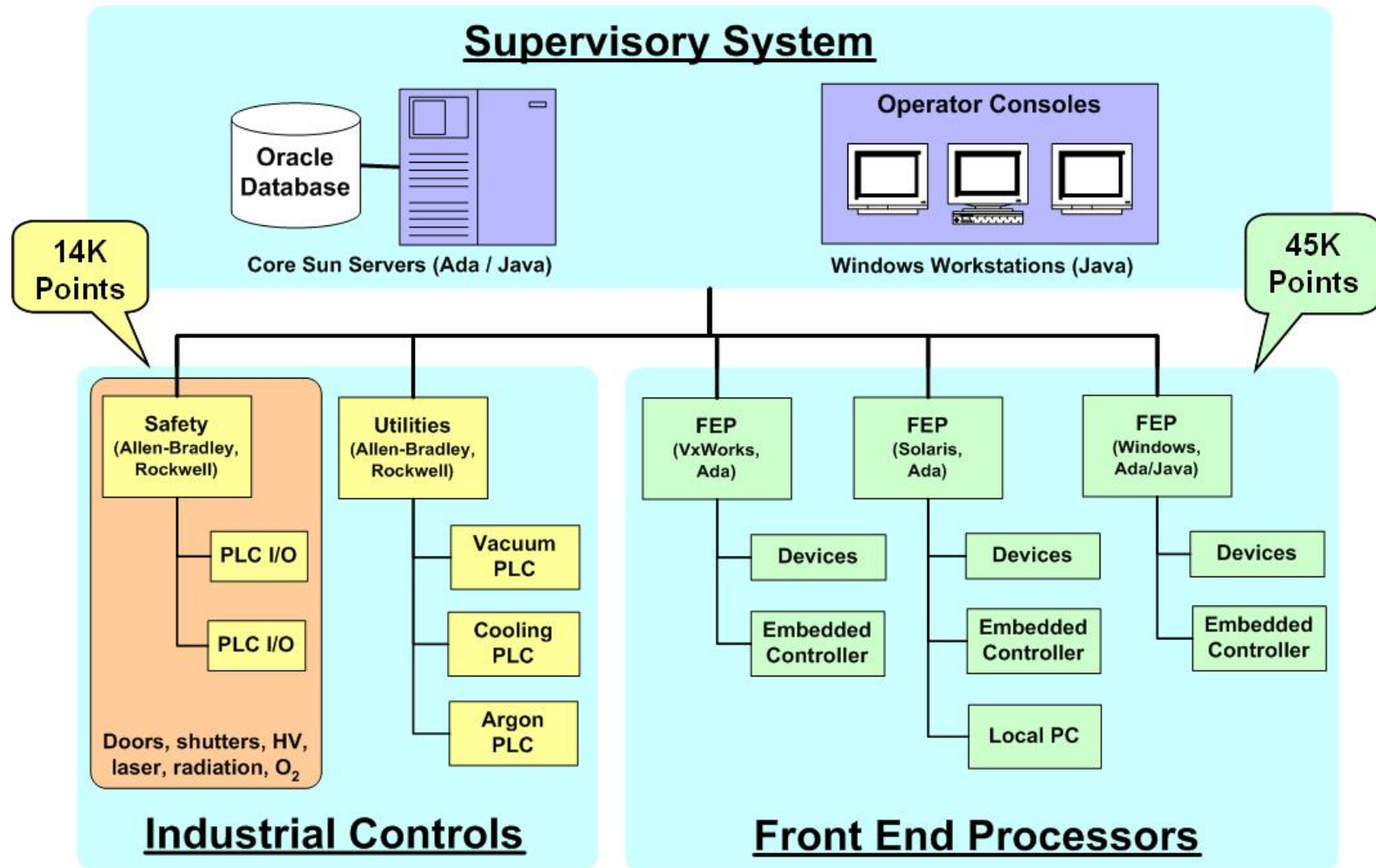
*The National Ignition Facility*

- **Software engineering**
  - Open distributed architecture based on industry tools
  - Formalized design process
  - Configuration Management
  - Product Integration
  - Formal Verification Testing
- **Computer science**
  - Application frameworks developed in-house
  - Standardized coding techniques
  - Extensive peer reviews

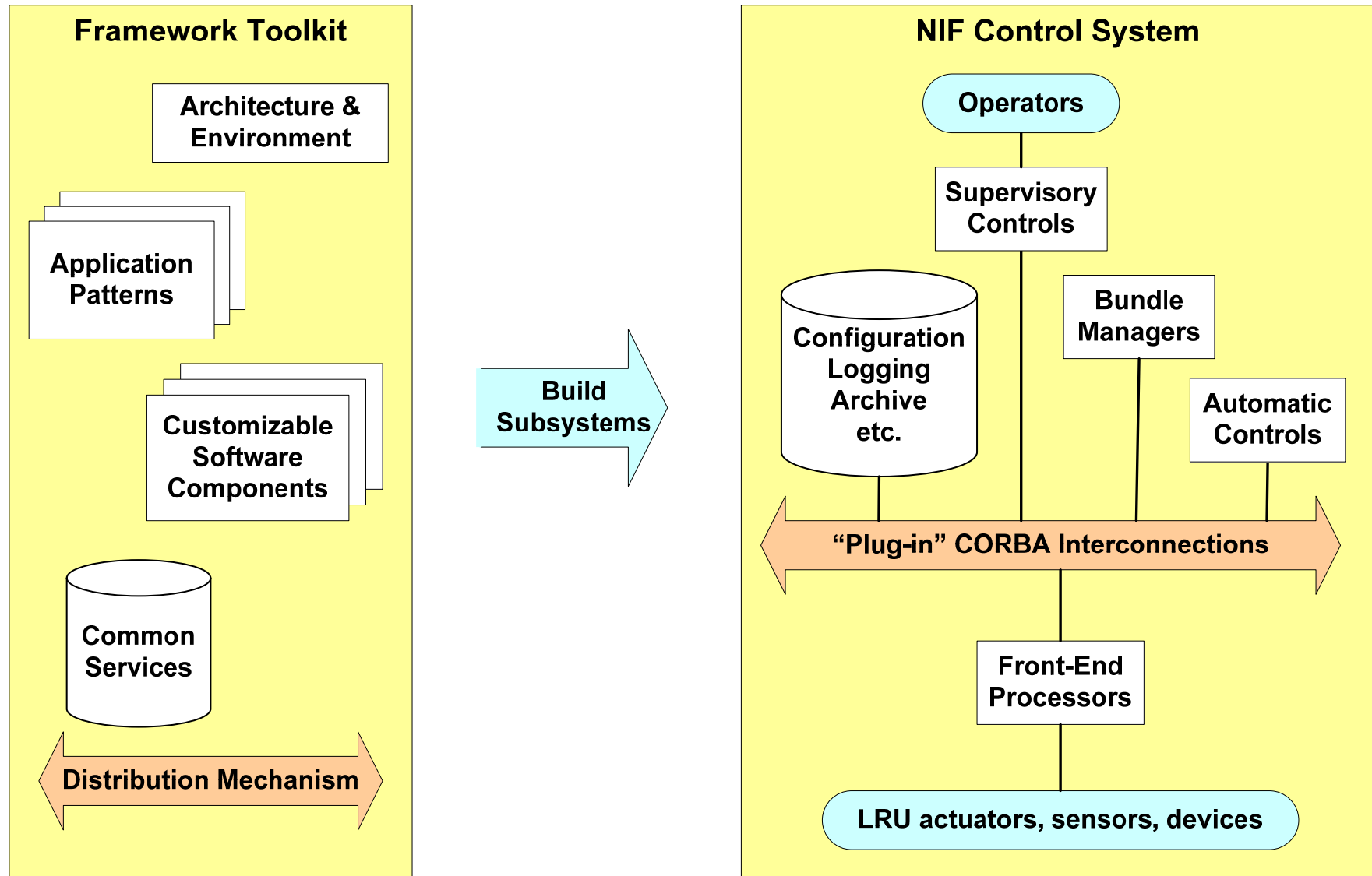
**Goal: Build team with the necessary experience and engineering rigor**



# A segmented and layered architecture was implemented to decompose the control system



# ICCS software framework is an object-oriented toolkit to build large-scale control systems





# Open software architecture supports multiple environments and simplifies technology evolution

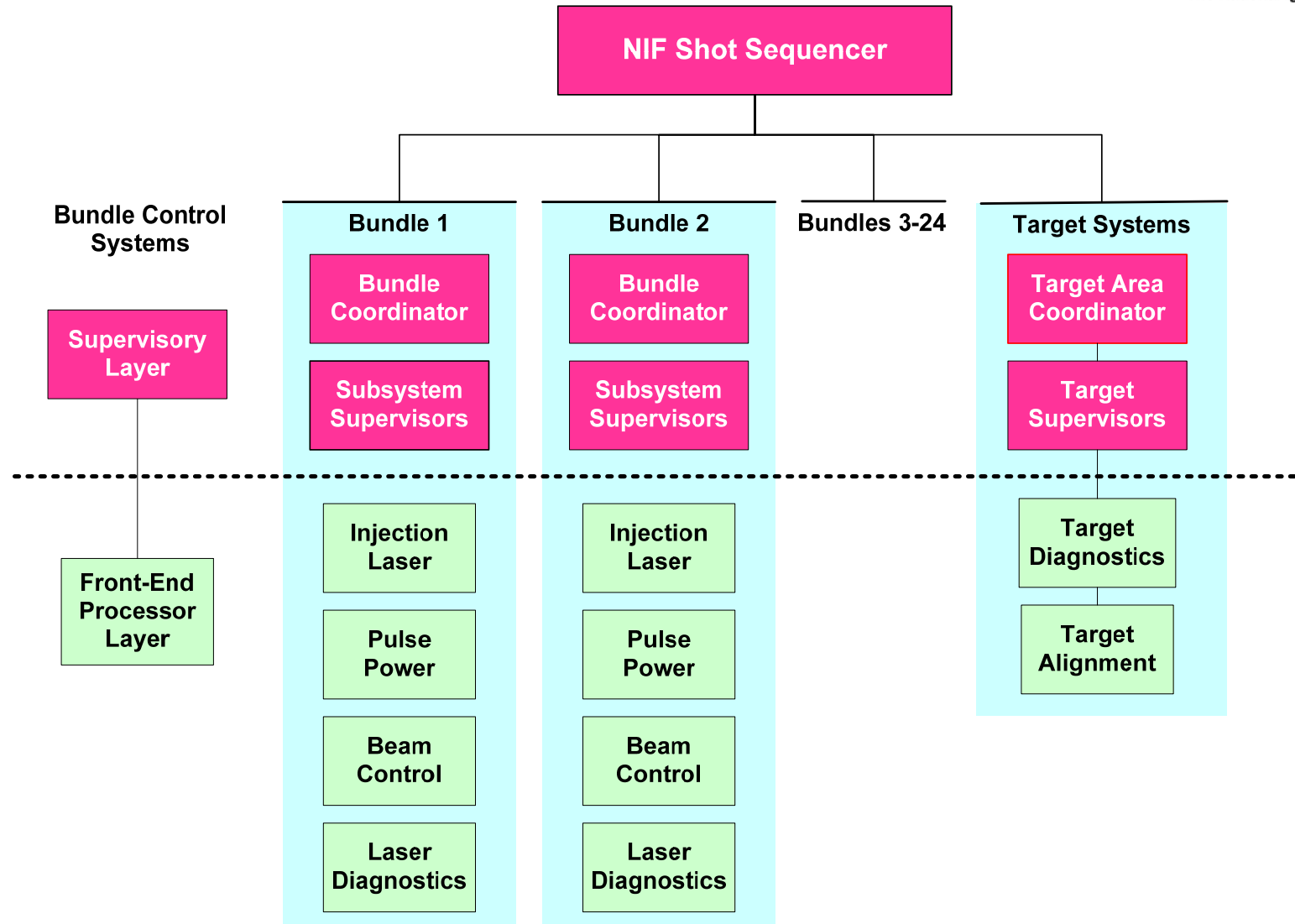
Category	Tool / Environment	Usage
Computers / Operating System	<ul style="list-style-type: none"> <li>- Sun (Solaris)</li> <li>- PowerPC VME (VxWorks)</li> <li>- X86 (WinXP, LINUX)</li> <li>- X86 PC104 (WinCE)</li> </ul>	<ul style="list-style-type: none"> <li>- Servers, FEPs w/o hardware</li> <li>- FEPs with hardware</li> <li>- Consoles, image processing</li> <li>- Target diagnostic controllers</li> </ul>
Languages	<ul style="list-style-type: none"> <li>- Java</li> <li>- Ada</li> <li>- C</li> <li>- IDL</li> <li>- XML</li> </ul>	<ul style="list-style-type: none"> <li>- GUIs, frameworks, commissioning tools</li> <li>- FEPs, servers and supervisors</li> <li>- Embedded controllers</li> <li>- On-line data and image analysis</li> <li>- Workflow models and scripts</li> </ul>
CORBA Distribution	<ul style="list-style-type: none"> <li>- JacORB</li> <li>- ORBexpress</li> </ul>	<ul style="list-style-type: none"> <li>- Java applications (open source)</li> <li>- Ada applications</li> </ul>
Software Development	<ul style="list-style-type: none"> <li>- Eclipse</li> <li>- IBM Rational</li> <li>- Ada Core Technology</li> <li>- ILOG</li> <li>- UML</li> </ul>	<ul style="list-style-type: none"> <li>- Java IDE (open source)</li> <li>- Solaris, VXworks Ada IDE</li> <li>- Windows, LINUX Ada IDE</li> <li>- Graphics builder tools</li> <li>- Object modeling</li> </ul>
Database	<ul style="list-style-type: none"> <li>- Oracle</li> </ul>	<ul style="list-style-type: none"> <li>- Data-driven configuration, archives</li> </ul>

**As technology evolves, CORBA allows ICCS to migrate with it**

# ICCS architecture permits each bundle to be built and commissioned independently of the others



The National Ignition Facility



**A partitioned architecture helped ICCS eliminate scaling risks**



# Data-driven architecture adapts quickly in the field to suit operating conditions

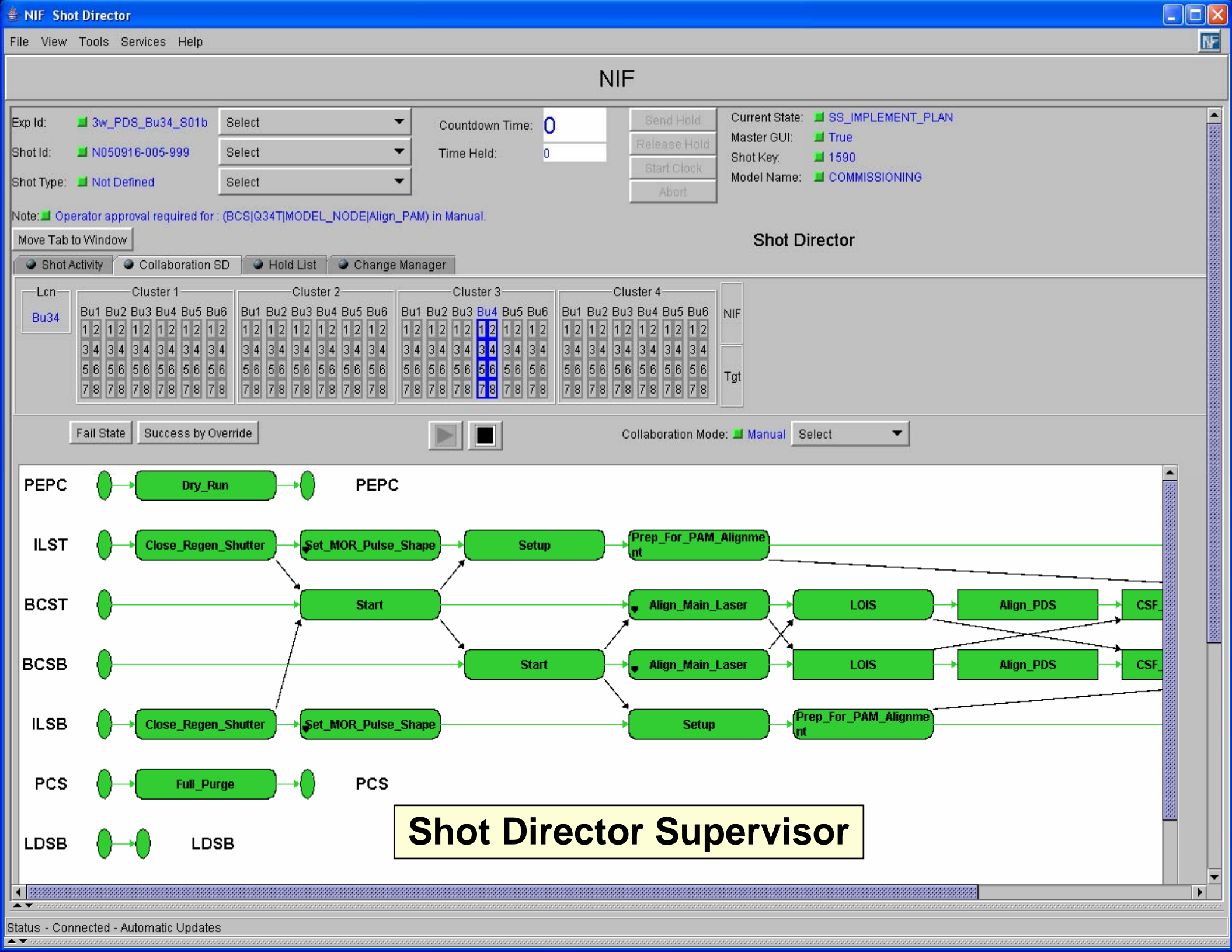


The National Ignition Facility


- **Flexible and responsive**
  - Nominal values deployed along with new capabilities
  - Data modified as components are commissioned
  - Operators update equipment calibration
- **Separate database instances map to the environment**

<b>Development:</b>	<b>Schema and initial data</b>
<b>Integration:</b>	<b>Validates data and schema</b>
<b>Formal Test:</b>	<b>Final verification</b>
<b>Production:</b>	<b>On-line use</b>
<b>Aux. Facilities:</b>	<b>Factory and maintenance facility support</b>

**Software modeling and data-driven automation behavior supports flexible point-of-use commissioning and optimization**







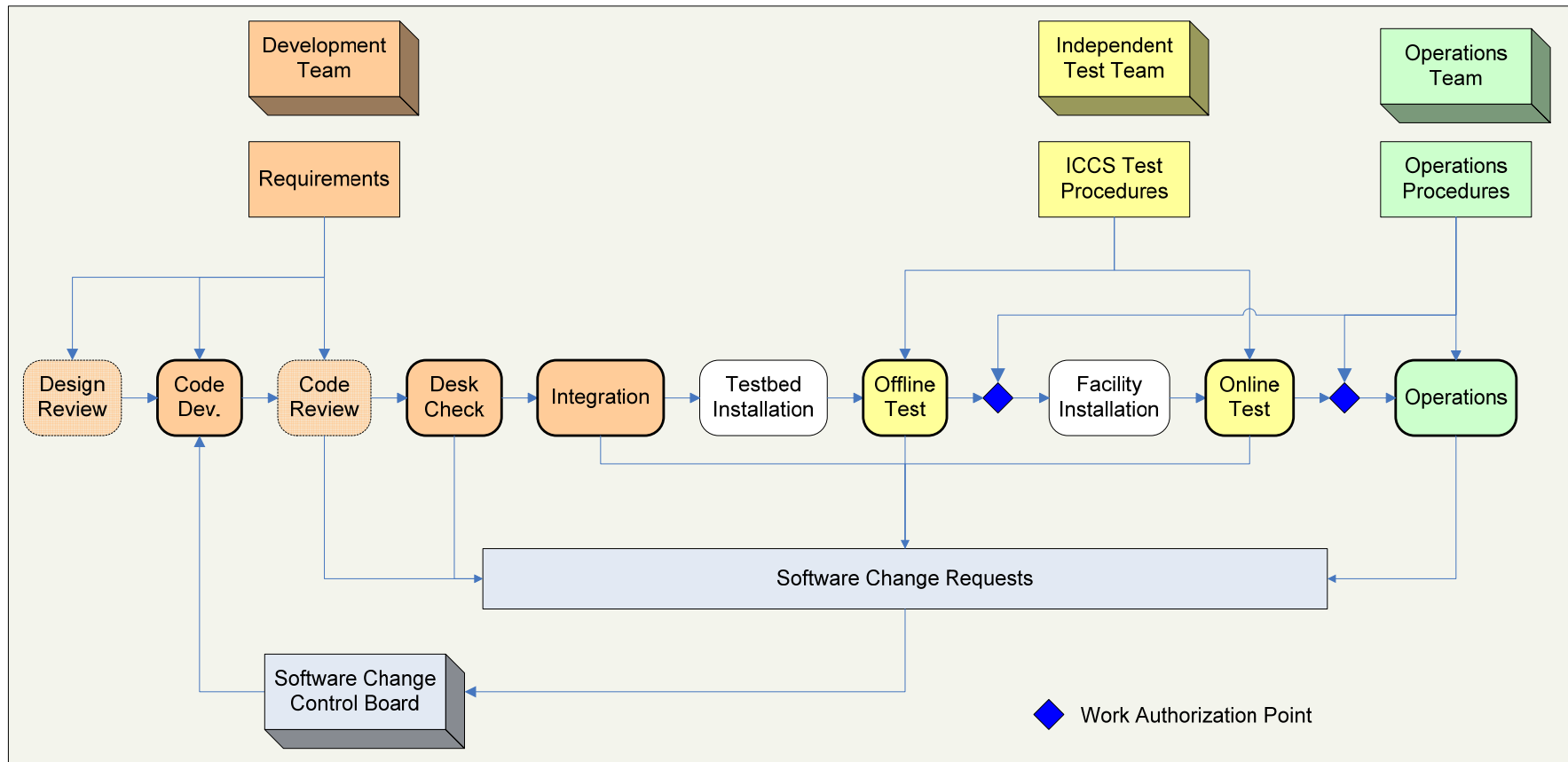
**Laser bay automated  
bundle shot controls  
delivered August 2005**

**Model-based automated shot control in routine use for  
commissioning and laser performance tests**

# Rigorous quality controls assure delivery of reliable software to the facility



The National Ignition Facility



**The QC process resolves software problems early, with more than 90% of problems found prior to commissioning or shot operations**

# **Incremental development strategy delivered functionality, while also iteratively managing risk**



*The National Ignition Facility*

- **Target high-level project milestones**
  - Major release planned every 12 months
  - Over 60 minor releases and patches deployed per year
- **Release content driven by**
  - Project goals and identified capabilities
  - Defect resolution
  - Improvements requested by operation staff
  - Framework enhancements
  - Perceived risks
- **Strict development lifecycle followed**
  - Managers plan release and collect detailed requirements
  - Developers design, code, unit test and integrate
  - Separate test team performs offline and online tests

**Goal: Support the overall project schedule while maintaining a manageable release content**



# Software metrics demonstrate the team's adaptability to deliver flexible releases



The National Ignition Facility

Release Type	Average Changes	Deployments per Year	Average Development Time	Average Integration Time	Average Test Time
Major	540	1	1 y	1 m	1.5 m
Minor	469	3	12 w	2.5 w	4 w
Service Pack	57	11	4 w	1 w	2.5 w
Patch	7	50	4 d	<1 d	2 d
Script & Database	Several per Day	Hundreds	NA	NA	NA

# **Releases are actively managed by closely monitoring the change tracking system**



*The National Ignition Facility*

## **Tracking System Database**

- Releases contain subsystems that are individually tracked
- Subsystem leads estimate the work in hours
- Change states tracked:
  - Assigned, Completed, Desk-checked, Integrated, Released

## **Active Management**

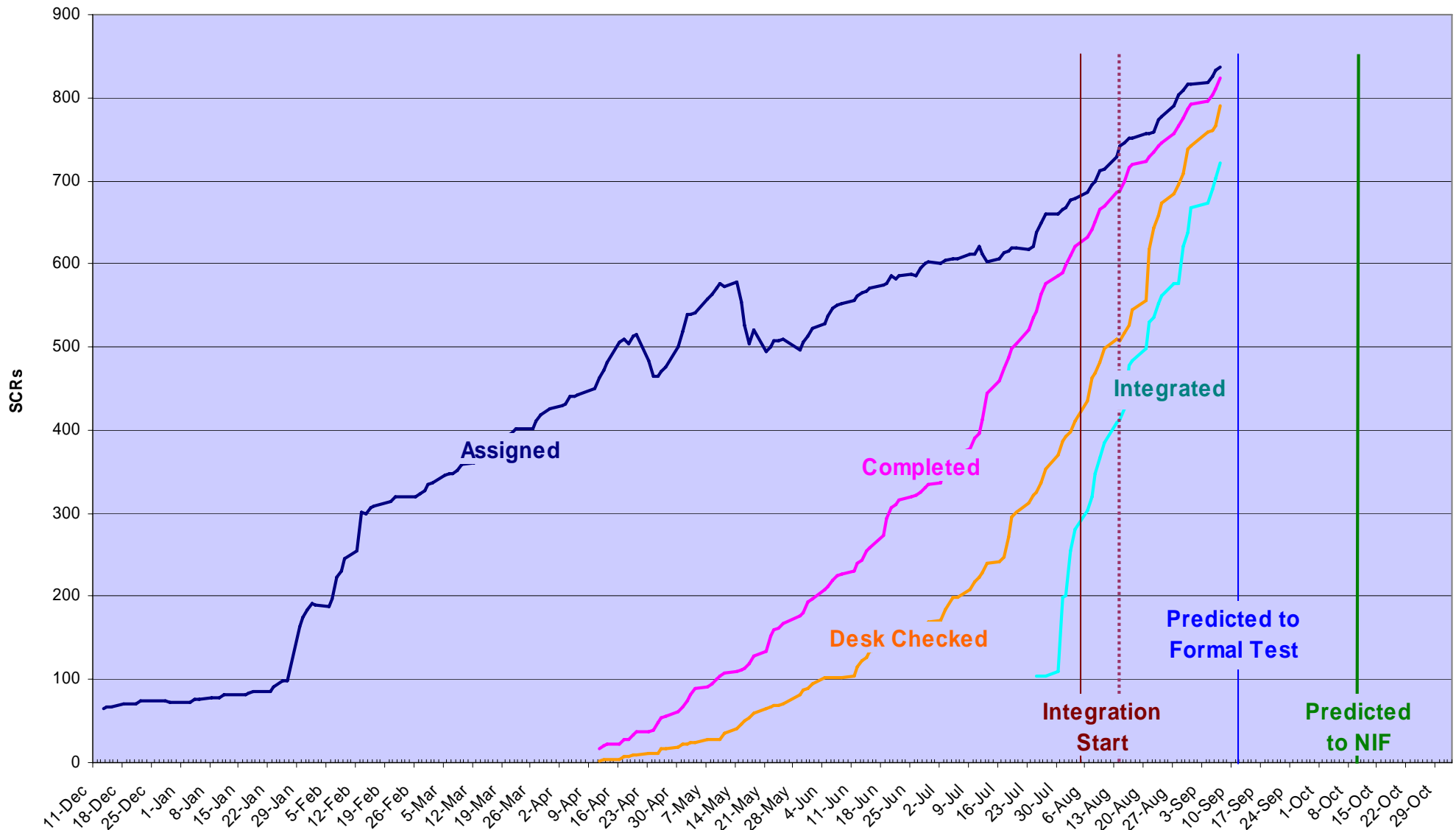
- Planning
  - Work assigned to each subsystem is analyzed
  - High levels of change requests indicate increased testing
  - Unrealistic developer workloads are replanned
- Release
  - Progress is tracked on a daily basis
  - Trends predict completion dates
  - Progress is easily understood
  - Release content continues to be adjusted as needed

**The tool maintains data on past release performance, which is used to flag potentially unreasonable expectations**

# Status of Software Change Requests (SCR) predicts the release completion date



The National Ignition Facility





# Peer reviews enforce design and coding standards during the development phase



The National Ignition Facility

- **Requirements Reviews**
    - Held with hardware and operations Subject Matter Experts
    - Assures developers and testers understand work
  - **Design Reviews**
    - Mandatory for new systems, or for complex changes
  - **Code Reviews**
    - Desk Checks performed by another developer
      - Does code meet requirements?
      - Does code implement the design properly?
    - Walkthrough Inspections
      - Team review of critical code or key interfaces
- NIF's 100% Desk Check policy catches 15% of total defects

**Goal: Find defects earlier in the process, preferably before testing when cost to repair is lowest**

# Efficient developer testing is a combination of unit testing followed by formal product integration

---



*The National Ignition Facility*

- **Unit Tests confirm code changes**
  - Informal peer review
  - Developed to specific standards and practices
  - Automated tests are being developed
- **Integration Tests ensure products work together**
  - Tests interfaces
  - Verifies proposed database changes
  - Applies various test levels
    - Change verification
    - Regression checks
    - Shot sequences
    - Scaling and loading tests

# Software releases are formally integrated in an “extreme programming” mode

---



*The National Ignition Facility*

- **Led by the Product Integration Manager**
- **Integration is started before all coding is completed**
  - **Gets the team into test and fix mode**
- **Daily integration goals are published**
  - **Lower-level software functions verified first**
  - **Shot sequences evaluated last**
  - **Goals updated based on previous day's results**
- **Extreme programming works for NIF**
  - **All developers on call (via pagers)**
  - **Development code base used for integration**
  - **Defects fixed on the spot where possible**
  - **Integration defects carefully prioritized**

**Goal: Produce a final integrated product with good quality**



# Independent verification demonstrates readiness



- **Off-line Release Verification Testing**
  - **Verifies software for delivery to NIF**
    - **Critical software controls are validated**
    - **Issues reviewed with Responsible Individuals**
    - **All testing is documented**
  - **Performed in an environment similar to NIF**
    - **Mimics NIF hardware in a low-cost test environment**
  - **Initial operator training on new features**
  - **Operator checklists revised to conform to the new software**
- **On-line Deployment Verification**
  - **Verifies software performs correctly with NIF hardware**
  - **Verifies build configuration and production database changes**
  - **Final operator shot training**

**All software releases (and patches) undergo independent offline testing**

# Offline test bed uses real devices where practical, along with software emulators for scaling tests



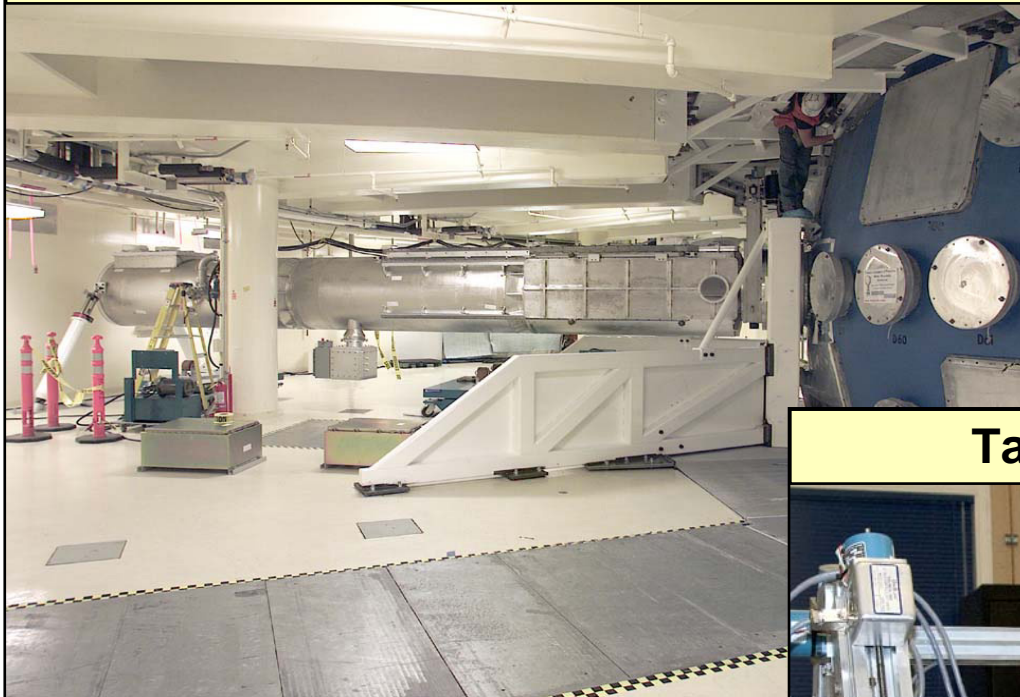


# Test bed simulators mimic operation of large-scale NIF Hardware

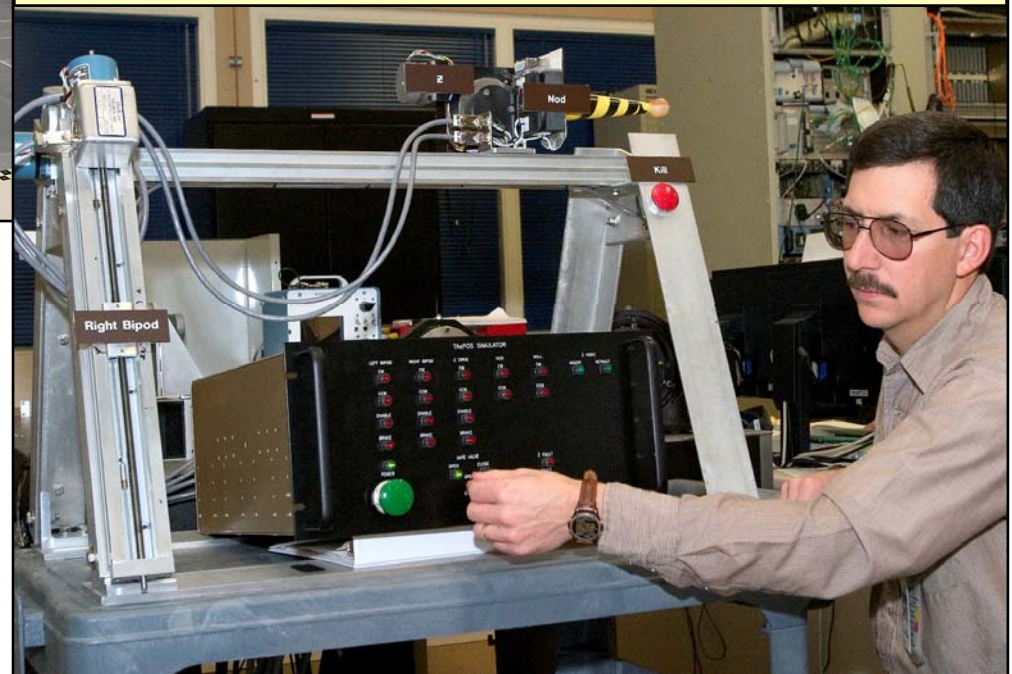


The National Ignition Facility

## Target Positioner in NIF



## Target Positioner Simulator





# Code configuration is managed by a dedicated staff of specialists

---



*The National Ignition Facility*

- **Code base managed across multiple environments**
  - Support mixed languages, development tools, and platforms
  - Several releases are simultaneously active
- **Automated tools validate build accuracy**
  - Automatic regression checker identifies back-revisions
  - Automatic platform checker identifies version mismatches across platform code bases
- **Online data/file changes are checked frequently to maintain conformity**
  - Online data changeable only through a formal permit and approval process

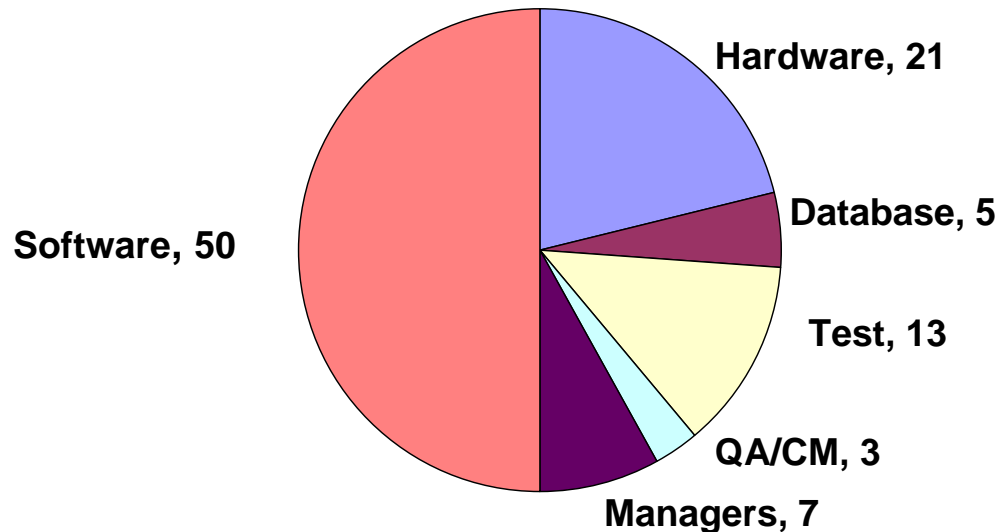
**Software and data configuration management are essential to produce predictable and traceable releases**

# The ICCS team includes diverse skills in software, controls, QC and laser technology



The National Ignition Facility

**ICCS Staff Composition, 100**



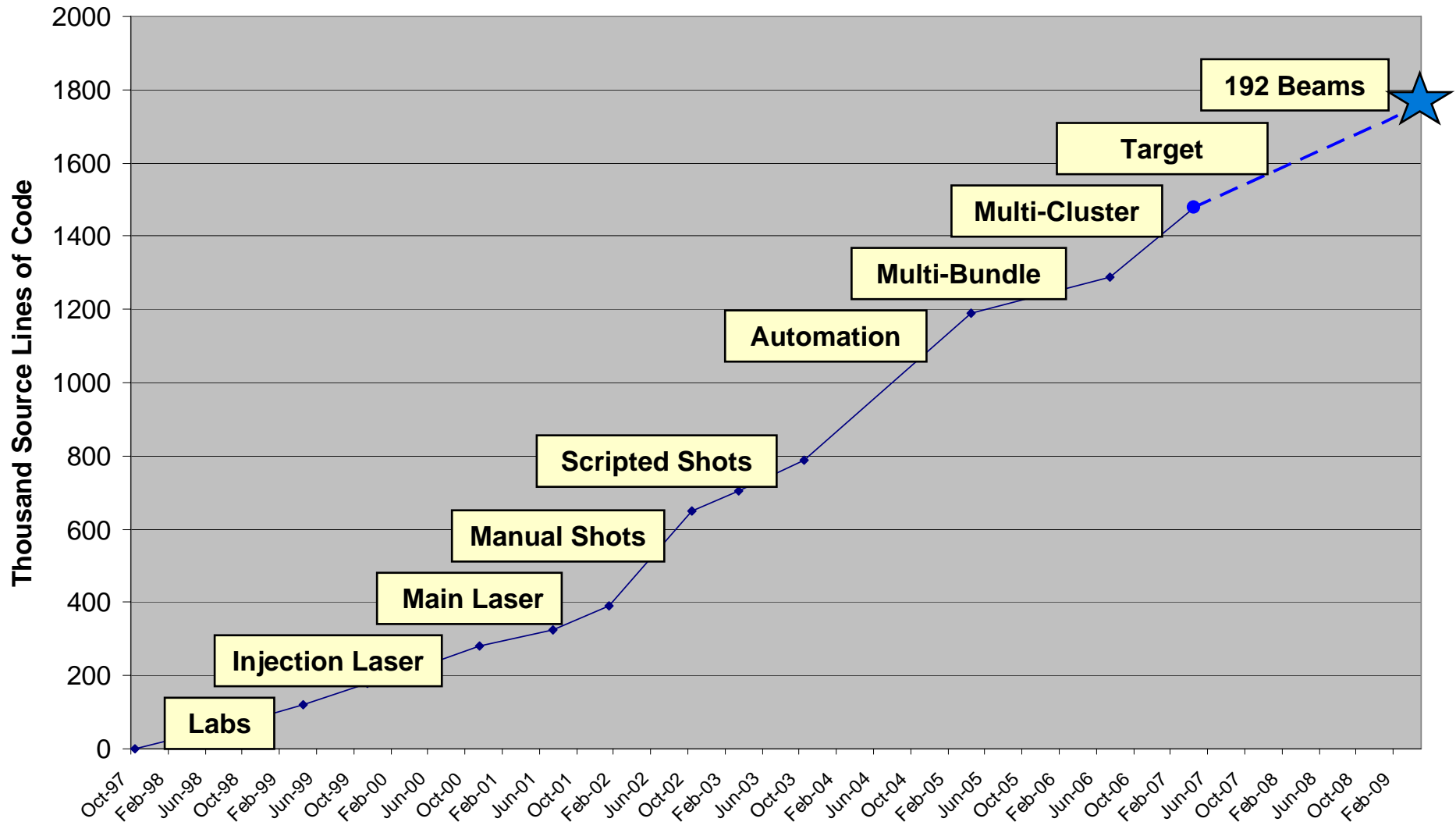
- Team averages 20 years in field
- Key experience drawn from other physics projects and relevant industries such as aerospace

- **Software skills**
  - Object-oriented
  - Database
  - User interfaces
  - Real-time
  - Image processing
  - Build management
  - Test engineering
  - Quality assurance
- **Hardware skills**
  - Motion controls
  - Timing systems
  - Data acquisition
  - Computers & networks
  - Manufacturing

# Software development of 1.8 million lines of code is greater than 85% complete



The National Ignition Facility



**ICCS successfully supported over 400 shots while the software was under construction**





**14 Main Laser Bundles  
Operationally  
Qualified  
October 2, 2007**

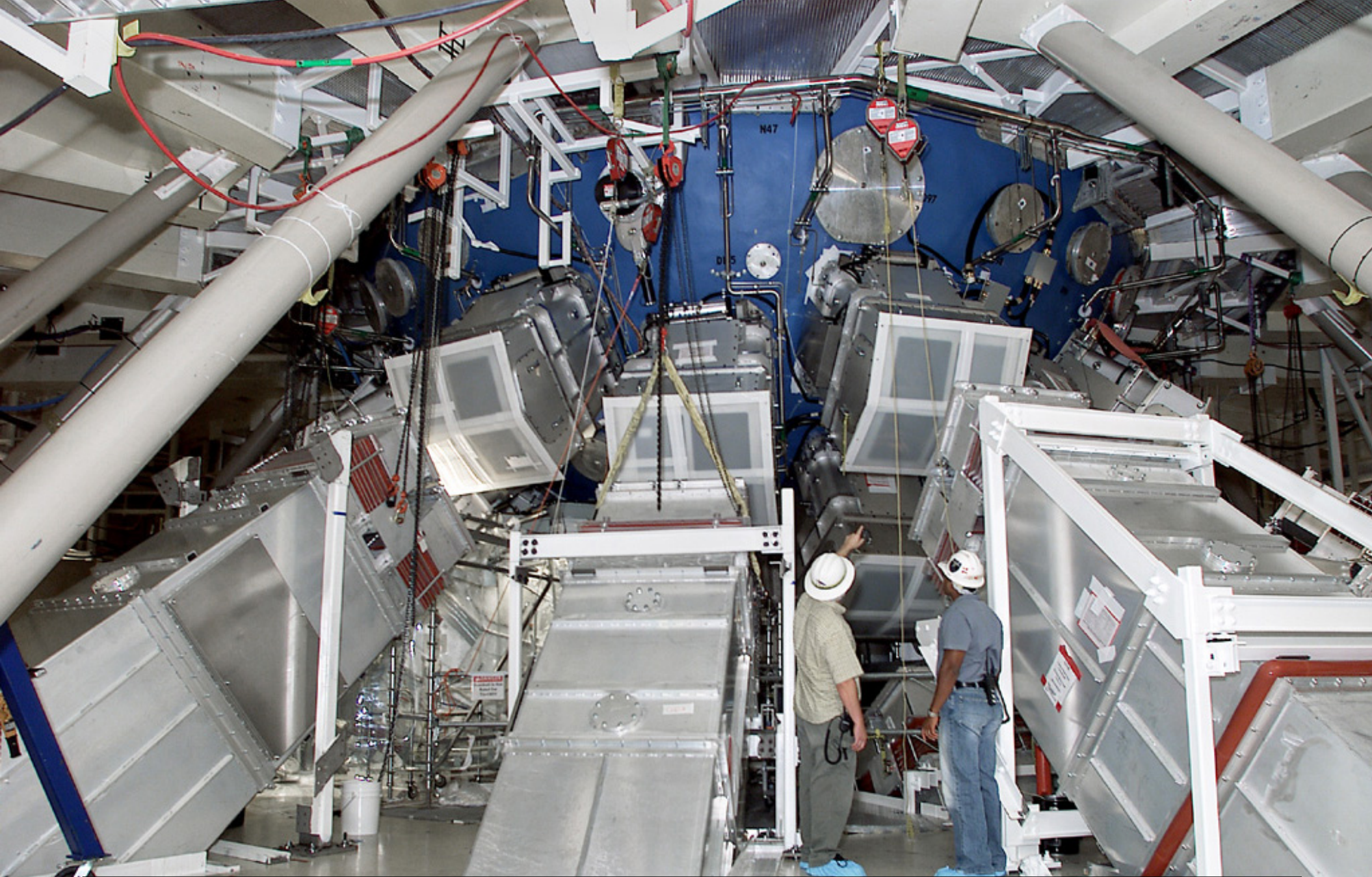
**World's Highest Energy  
Laser – 2.4 MJ  $1\omega$**



NIF-0807-1:  
31BVVW/sb

**The control system scales and operates all bundles in parallel**

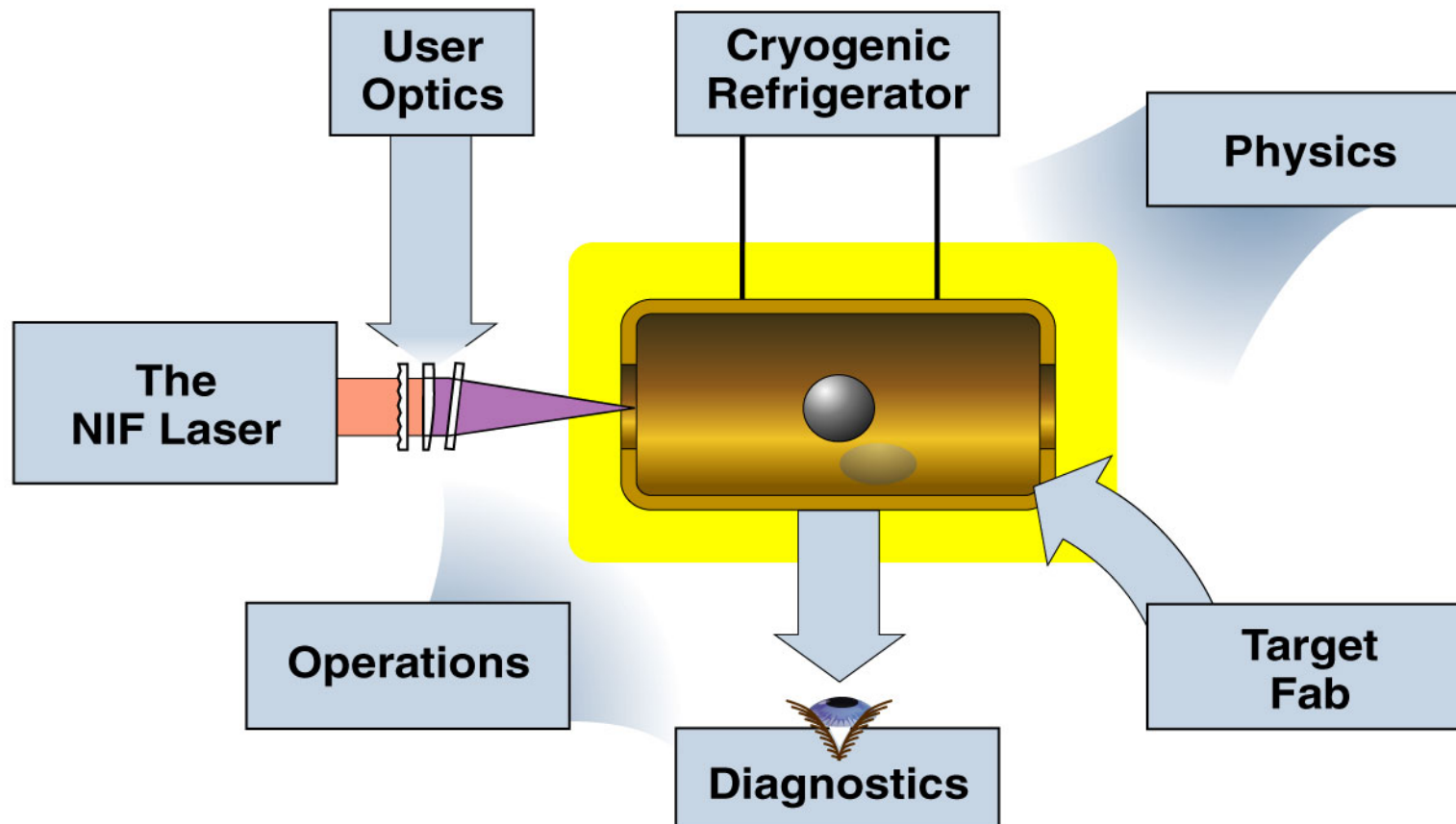




**The focus now is on automated shots with target-area systems including final optics, positioners and diagnostics**



# The next experimental challenge is a demonstration of ignition on NIF



**Our plan for 2009–2010 concentrates on systems integration and executing a credible ignition campaign**

# Large control systems can be developed on time with expectation of consistent reliability



*The National Ignition Facility*

- **Architecture**
  - Segment, partition, layer, and (open) distribution
  - Common frameworks, design patterns and code templates
  - Address constraints and scaling concerns
- **Development practices**
  - Careful planning under authorized change control
  - Configuration management of code base and data
  - Diverse, experienced, and specialized staff
  - Resolve risks as well as urgent issues
- **Quality controls**
  - Peer reviews and unit testing
  - Intensive product integration
  - Independent formal testing
  - Measure quality to guide corrective actions

**NIF continues on track for project completion in 2009, followed by National Ignition Campaign experiments beginning in 2010**



