CCT Lessons Learned
what we did,
why we did it, and
what we would do differently

John Ohlinger

6 February 1999
Agenda

- “Good Things”
- Lessons Learned
- Room for Improvement
Purpose

- What we did
  - what we did right
  - What we would do differently on the next program
- Hopefully you can learn from our experiences
"Good Things"

- Weekly program telecon status meetings
- Active action item tracking system
- Joint CPO/CCTC calendar
- Weekly CPO staff meetings
- Tracking lists for all activities
- CCT Readiness Management Plan (CRMP)
- Use of web, teleconferencing and email
- Running program in an unclassified environment
- Incremental development approach
- Teamwork
Lessons Learned

Categories

- Program Management
- Schedules/Metrics
- Communication
- Requirements
- Reuse Aspects
- Documentation
- Software/Tools
- Architecture/Design
- Testing
Program Management

- Utilize a single integrated management structure for all program elements
Schedules/Metrics

- Provide detailed Increment schedules well before Increment Design Reviews
- Reduce overlap of major reviews
Provide direct line of communication between contractor and CPO personnel
- Hold technical interchange meetings to resolve issues

Utilize IPT concepts - have buy in from contractor and government on goals
Requirements

- Develop detailed (not general) requirements to avoid late breaking design issues
- Write clear, concise, testable requirements
Reuse Aspects

- Reuse aspects were not as fully developed as the technical requirements
  - Standards for ease of use of documentation & design, adequacy of testing, processes, etc. were not given as much emphasis and consideration as technical requirements
Documentation

- Need clear understanding of reuser documentation (level of detail, audience, etc.)
- Include a technical writer on staff
Documentation RFC’s

- Identify clearly defined RFC/CM Process from the start (mutually agreed to by all parties)
- Set up standard email lists for all RFC recipients (eliminate Single Point-of-Failure)
Architecture/Design

- Have clear understanding of design artifact content expectations for each milestone
- Formulate a stable set of Architecture, Design and Construction Patterns early in the process
- Have clear understanding of how S/W interfaces & components are to be documented
Testing

- Do not limit test program to component-level testing
  - Include System Level Scenarios
  - Include Stress Testing
Testing

- Involve community as early as possible in test scenario development
- Provide detailed data as early as possible
  - Truth models, scenarios, algorithms, etc.
Room for Improvement

- Improve understanding of productivity tools (Word, Frame, Adobe Products, WinZip) - to produce uniform documents
- Utilize more sophisticated Internet tools (NetMeeting)
- Utilize Internet style “pagers” (icq.com) for quick resolution of issues
For more information

- John F. Ohlinger
  - ohlingej@NRO.mil
  - secure 850-3070
  - uncl 703-808-3070