Carnegie Mellon
Software Engineering Institute
Pittsburgh, PA 15213-3890

Specifying Initial Design Review (IDR) and Final Design Review (FDR) Criteria

Mary Ann Lapham

Sponsored by the U.S. Department of Defense

Outline

Introduction – Rationale for IDR and FDR

Definitions

Engineering Emphasis

IDR and FDR Criteria

Applying the Criteria

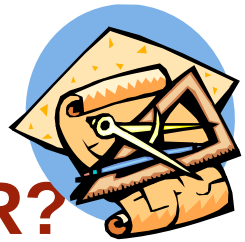
Conclusion

Introduction – Rationale for IDR/FDR? 1

Many DoD development programs initial capabilities are prototypes for proof of concept and lingering operational capability for extended evaluation and exploitation

System representation developed in Technology Development includes

- definition of HW and SW configuration items (CIs)
- how functionality is distributed across the CIs
- how they interact to provide system level capabilities



Introduction – Rationale for IDR/FDR?

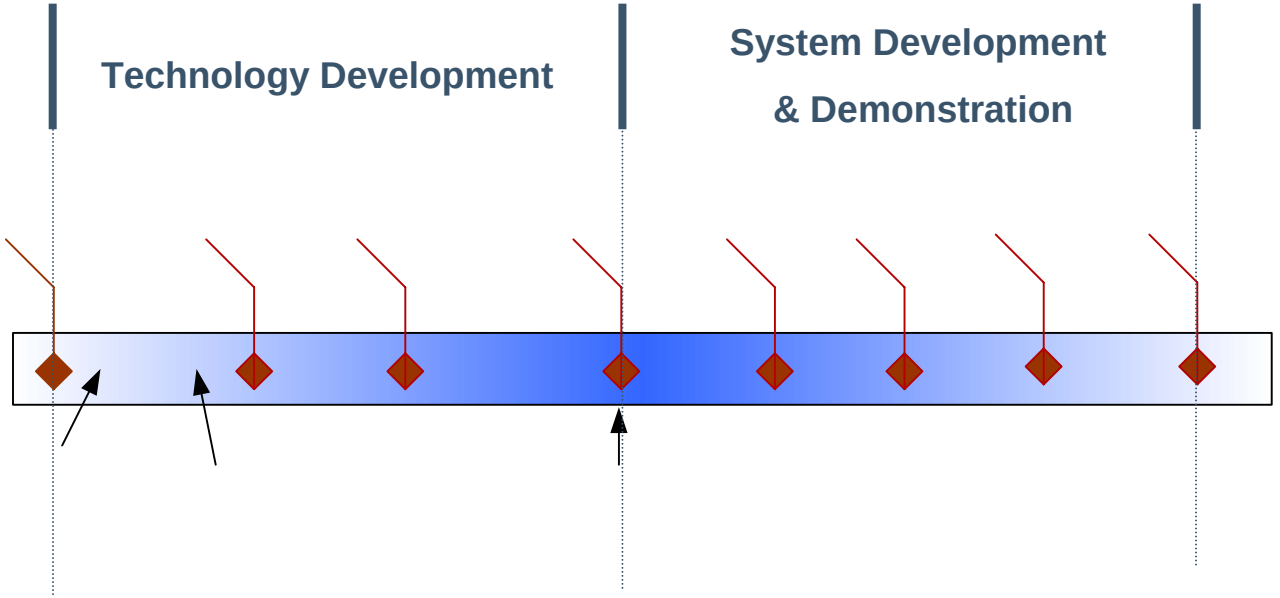
Need increasingly rigorous architectural reviews of system level capabilities to verify desired emergent properties of the system

- security
- flexibility
- extensibility
- reconfigurability
- interoperability
- and others as identified for particular system

Produce documentation of engineering and management areas

- “hard”, risk laden, requirements done early in lifecycle with effective risk mitigation plans in place
- comprehensive recording of design and decision rationale (tracking why and perhaps how)

IDR/FDR Context



Outline

Introduction – why have IDR and FDR

Definitions

Engineering Emphasis

IDR and FDR Criteria

Applying the Criteria

Conclusion

IDR/FDR Purpose *

Evaluate system and software architecture and design developed in Technology Development

Ensure desired/required capability is well represented during prototype demonstration

Ensure high risk items have been addressed to reduce/mitigate the risk as much as possible

Document engineering and management decisions

* Software focus for rest of briefing

IDR/FDR Definitions 1

Not explicitly defined in any regulations or policies

Defined by Program Office

Part of Technology Development phase

Roughly equivalent to PDR/CDR in System Development & Demonstration phase

Basis of developing additional criteria

- MIL-STD-1521B
- USAF Software Management Guidebook

IDR/FDR Definitions 2

Working definitions:

IDR – formal technical review of prototype design approach for a CI or CSCI including evaluation of progress, technical adequacy, and risk resolution on a technical, cost and schedule basis.

FDR – formal technical review of prototype detailed design approach for a CI or CSCI including evaluation of progress, technical adequacy, and risk resolution on a technical, cost and schedule basis.

Word of Caution

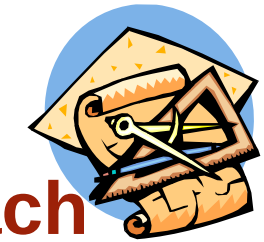
MIL-STD-1521B assumed waterfall approach

Today's development environments often incorporate incremental or spiral development approaches

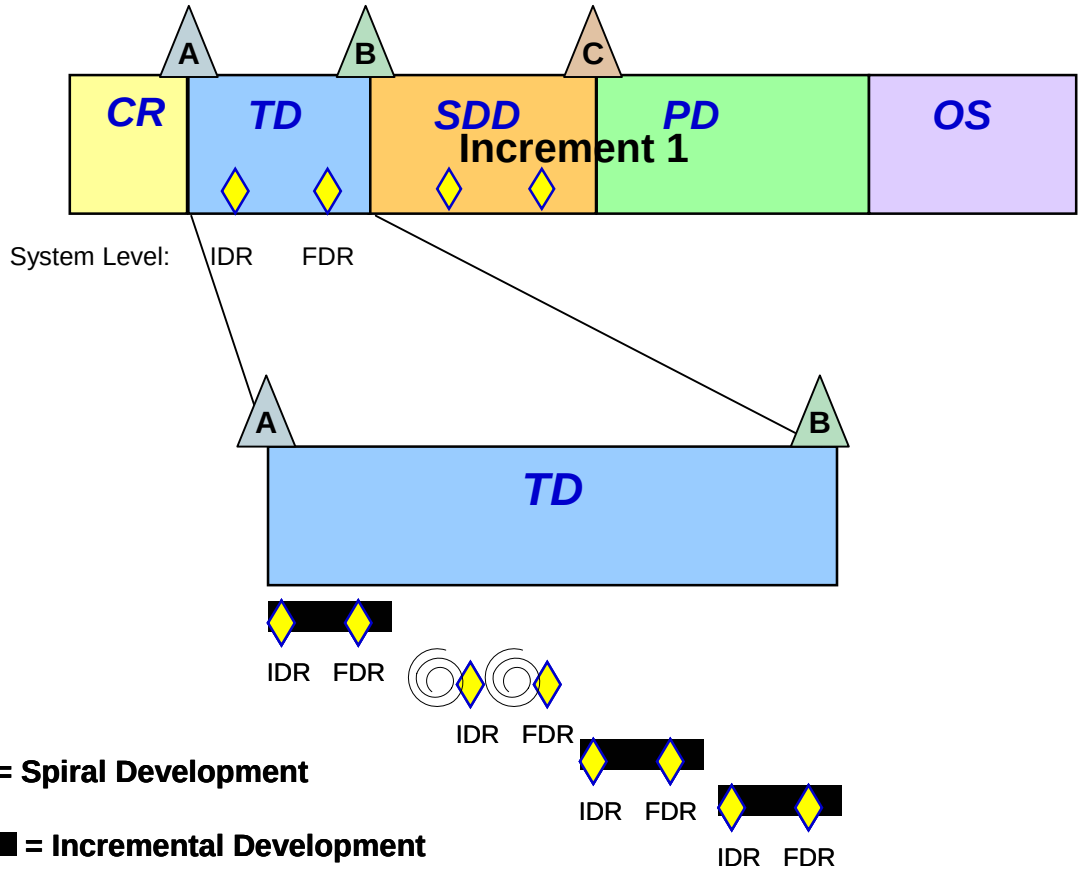
With incremental or spiral approach, functional decomposition occurs in increments or as program evolves during spirals

- Thus, multiple IDRs and FDRs possible
- Could have combination of approaches

Keep in mind to take development environment into account when applying the proposed IDR/FDR criteria



Example Incremental/Spiral Approach



Outline

Introduction – why have IDR and FDR

Definitions

Engineering Emphasis

IDR and FDR Criteria

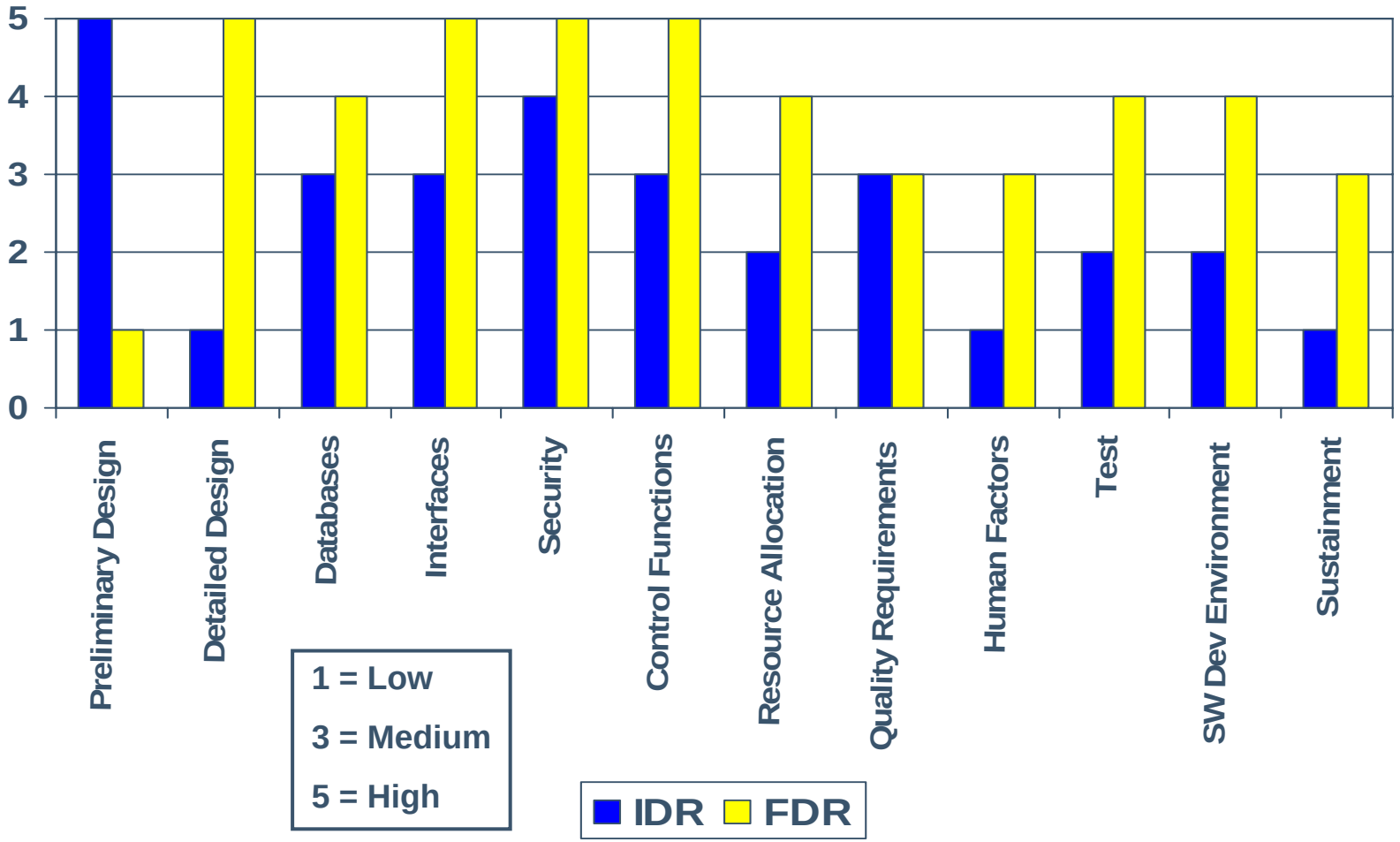
Applying the Criteria

Conclusion

12 Software Development Areas

1. Preliminary Design
2. Detailed Design
3. Databases
4. Interfaces (internal and external)
5. Security (information assurance specific to software)
6. Control Functions (high-level description of executive control and start/recovery features)
7. Resource Allocation (allocation across all CSCIs including timing, sequencing, and relevant constraints)
8. Quality Requirements (reliability, maintainability, supportability, producibility, safety, extensibility, flexibility, reconfigurability, and interoperability)
9. Human Factors (includes natural environment)
10. Test (all types)
11. Software Development Environment (tools, development approach, and personnel)
12. Sustainment (support resources, software test equipment, and technical manuals)

Engineering Emphasis



Outline

Introduction – why have IDR and FDR

Definitions

Engineering Emphasis

IDR and FDR Criteria

Applying the Criteria

Conclusion

IDR/FDR Pre- and Post- conditions

Define a set of conditions for entry and exit from IDR and FDR

Criteria demonstrable conditions that require something be
- Created
- Available
- Accomplished

Artifacts required for each condition
- Potential UML artifacts provided
- Other documentation as needed
- Sometime use Model Analysis (analysis of UML artifacts and supporting documentation)

IDR Software Pre-Conditions 1

Pre-Conditions	Potential UML Artifacts
System/Subsystem functional and performance requirements baseline Compatibility between CSCI and CIs Baselined interchangeability /replaceability decisions	• Scenarios and Use Cases • Component diagrams • Deployment diagrams • Sequence diagrams • Activity diagrams + Model analysis
Preliminary system and CSCI architecture	Component Diagram
Preliminary system software design accommodated by architecture	Model analysis

IDR Software Pre-Conditions 2

Pre-Conditions	Potential UML Artifacts
Make/buy decisions (legacy and COTS)	
Functional performance interface requirements (high level)	• Use cases
Design implementation trade studies	• Model analysis if alternatives modeled in UML
Sub-level IDRs completed	
ECPs including CSCI impacts	
Software increments planned and defined including allocated requirements	

Note: Blank cell means no applicable potential UML artifact. UML may be insufficient to document all aspects.

IDR Software Post-Conditions

Post Conditions	Potential UML Artifacts
Software risk management process defined and implemented	
Software architectural level design established	Model analysis
System/Software Engineering Environment defined and controlled	
Preliminary software design is defined and documented and satisfies functional and performance requirements	Model Analysis
Software increments defined	
Following items defined and documented • Interface Control Document • Software Metrics • Software Test Plan • COTS and reuseable software • Software development process • Software estimates	
Life-cycle software support requirements update	
Software item approved to start detailed design	

Note: Blank cell means no applicable potential UML artifact. UML may be insufficient to document all aspects.

FDR Software Pre-Conditions

Pre-Conditions	Potential UML Artifacts
Software requirements satisfy system/subsystem requirements baselines	Use Case diagrams and specifications
Software increments planned and defined including allocated requirements	
Software architectural level design established	Class diagrams and deployment diagrams
Requirements are satisfied by the design architecture and approach	Sequence diagrams
Complete detail level designs and specifications	Class diagrams, sequence diagrams, state charts

Note: Blank cell means no applicable potential UML artifact. UML may be insufficient to document all aspects

Carnegie Mellon
Software Engineering Institute

FDR Software Post-Conditions

Post-Conditions	Potential UML Artifacts
Integrated software development toolset is implemented and ready to support code and unit test	
Detailed designs reviewed and approved for implementation	
Metrics program in place	
Test descriptions complete	Sequence diagrams Deployment diagrams
Interface descriptions complete and in baseline	Class diagrams Sequence diagrams State Charts
Development files established and maintained	
CSCI design approved for start code and unit test	Class diagrams Sequence diagrams State Charts

Note: Blank cell means no applicable potential UML artifact. UML may be insufficient to document all aspects

IDR and FDR Criteria Summaries

Based on 12 areas of software development

Not all 12 areas represented in criteria (example: detailed design in IDR and preliminary design in FDR)

Potential UML artifacts provided

Note that blank cells indicate that an applicable UML artifact for that area does not exist

Summary IDR SW Evaluations 1

Areas of Concern	Potential UML Artifact
CSCI Preliminary Design • Functional Flow • CSCI Structure • Interfaces • COTS / GOTS/Reuse	• Sequence or Swim Lane Diagrams • Class Diagrams where classes are CIs stereotyped with kind of CI (HW or SW) • Deployment Diagrams (SW allocation to HW)
CSCI Security (Information Assurance (IA))	
CSCI Control Functions	Interaction overview diagram (activity diagram variant)
CSCI Resources Allocation	Profile for Schedulability, Performance, and Time
Quality Requirements • Reliability • Maintainability • Supportability • Producibility • Safety • Extensibility • Flexibility • Reconfigurability • Interoperability	Model Analysis

Note: Blank cell means no applicable potential UML artifact. UML may be insufficient to document all aspects

Summary IDR SW Evaluations 2

Areas of Concern	Potential UML Artifact
Human Factors including natural environment	• Use cases • Sequence diagrams • Activity diagrams
Test	• Deployment Diagram
Changes to Software Development Environment • CSCI Tools • Development Approach • Test • Personnel	
Sustainment • Support Resources • Software Test Equipment • Technical Manuals	

Note: Blank cell means no applicable potential UML artifact. UML may be insufficient to document all aspects

Summary FDR Software Evaluations

Areas of Concern	Potential UML Artifact
Software Detailed Design	• Class diagrams • Sequence diagrams • State Charts
Database Design	• Class diagrams
Interface Design	• Sequence diagrams • State Charts
Quality Requirements	• Annotated class diagrams • Annotated sequence diagrams
Sustainment • Maintenance /Maintenance Data • Support Resources • Software Test Equipment • Technical Manuals	
Human Factors	
Test	• Use Case diagrams • Use Case specifications • Sequence diagrams

Note: Blank cell means no applicable potential UML artifact. UML may be insufficient to document all aspects

Detailed IDR/FDR Criteria Examples

Information includes

- Software development area of concern
- Suggested detail/action
- Potential UML artifacts

IDR Software Evaluations 1

Areas of Concern	Suggested Detail/Action	Potential UML Artifact
CSCI Preliminary Design • Functional Flow • CSCI Structure • Interfaces • COTS / GOTS/Reuse	• Top level description of the functional flow for all software TRD requirements including interfaces • Architectural view(s) of the top-level structure with explanatory text including reasons for choosing the components, the development methodology, and support programs. • Interfaces both internal and external meet TRD specifications (defined, potential list of data, top level data dictionary) • Provide description and characteristics of COTS • Use of approved design methodology • Human Factors Engineering principles used in design	• Sequence or Swim Lane Diagrams • Class Diagrams where classes are CIs stereotyped with kind of CI (HW or SW) • Deployment Diagrams (SW allocation to HW)

FDR Software Evaluations 1

Areas of Concern	Suggested Detail/Action	Potential UML Artifact
Software Detailed Design	• Establish integrity of software design • Identify additional in-progress reviews as needed - Action items - ECP modifications - Sizing and timing data updates - Testing results • Supporting documentation for each design - Criteria and design rules for requirement allocation to lower level units - Information flow between lower level units - Design details – timing, sizing, data definitions, data storage and allocations • System allocation (architecture view) • Review SDD, System and Subsystem specifications, Test Plan, and Software Product Spec • Progress on CSCI IDR action items • Schedule for remaining milestones • Identification of outstanding risk and mitigation plans • Update since last review of all delivered software • Detailed design characteristics of all interfaces	• Class diagrams • Sequence diagrams • State Charts

Software Evaluations – the rest

Due to time constraints, other areas are located in backup slides

Outline

Introduction – why have IDR and FDR

Definitions

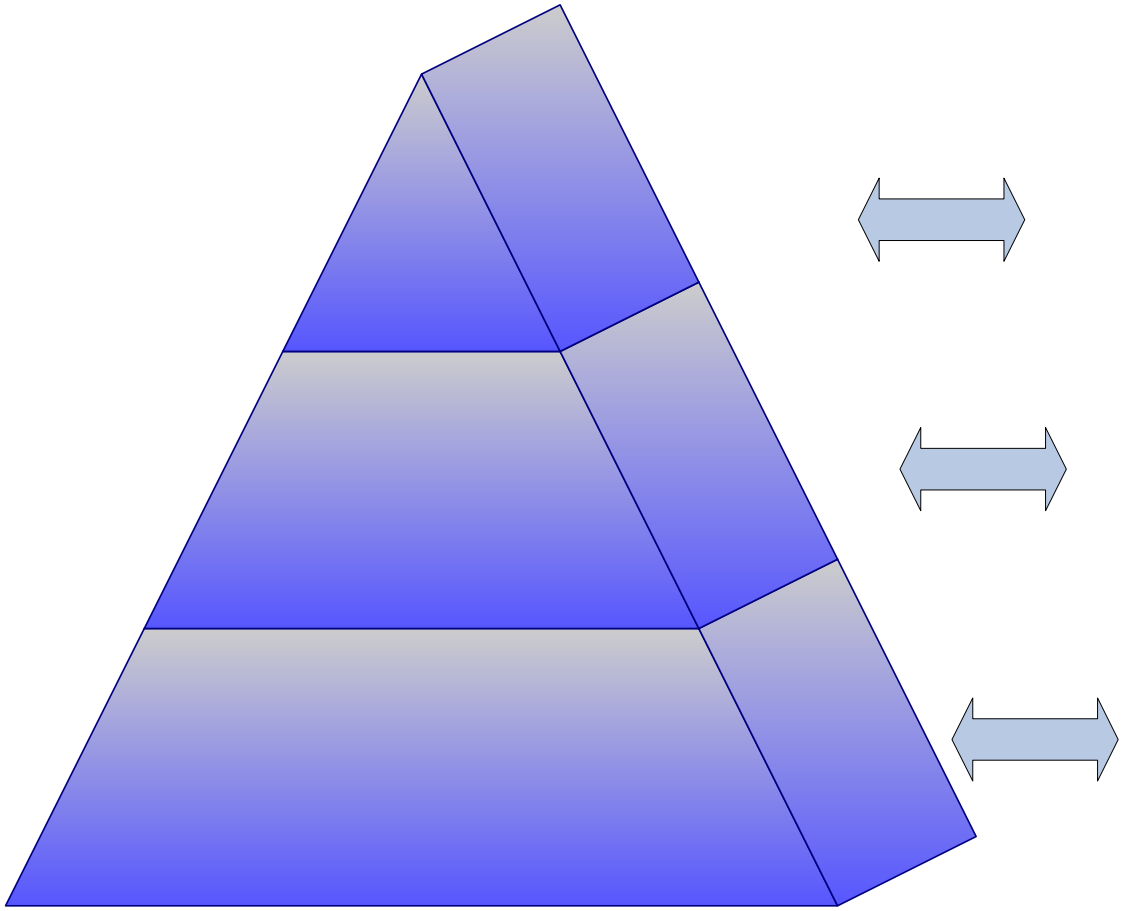
Engineering Emphasis

IDR and FDR Criteria

Applying the Criteria

Conclusion

Product Validation Hierarchy



Outline

Introduction – why have IDR and FDR

Definitions

Engineering Emphasis

IDR and FDR Criteria

Applying the Criteria

Conclusion

Conclusion

IDR and FDR during Technology Development can ensure

- Required capability is demonstrated
- High risks have been mitigated

Provide a way to introduce rigor and formality ensuring a required level of progress, technical adequacy, and risk resolution on a technical, cost and schedule basis

Adapt criteria to your program's environment and needs

- UML artifacts
- Non-UML artifacts
- Guidance to contractors
- Add IDR/FDR to Integrated Master Plan and Integrated Master Schedule

Carnegie Mellon
Software Engineering Institute

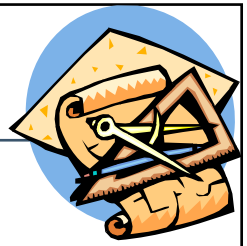
Questions

Contact me:

Mary Ann Lapham
Carnegie Mellon
Software Engineering Institute
4500 Fifth Avenue
Pittsburgh, PA 15312

412-268-5498
mlapham@sei.cmu.edu

Presentation based on Technical Note (Specifying Initial Design Review (IDR) and Final Design Review (FDR) Criteria) located at:
www.sei.cmu.edu/publications/documents/06.reports/06tn023.html



Backup Slides

Sponsored by the U.S. Department of Defense

PDR Definition

"Preliminary Design Review (PDR). This review shall be conducted for each configuration item or aggregate of configuration items to (1) evaluate the progress, technical adequacy, and risk resolution (on a technical, cost, and schedule basis) of the selected design approach, (2) determine its compatibility with performance and engineering specialty requirements of the Hardware configuration Item (HWCI) development specification, (3) evaluate the degree of definition and assess the technical risk associated with the selected manufacturing methods/processes, and (4) establish the existence and compatibility of the physical and functional interfaces among the configuration item and other items of equipment, facilities, computer software, and personnel. For CSCIs, this review will focus on: (1) the evaluation of the progress, consistency, and technical adequacy of the selected top-down design and test approach, (2) compatibility between software requirements and preliminary design, and (3) on the preliminary version of the operation and support documents."

Reference: DoD, MIL-STD-1521B, Military Standards, Technical Reviews and Audits for Systems, Equipments, and Computer Software, 4 June 1985.

CDR Definition

"Critical Design Review (CDR). This review shall be conducted for each configuration item when detail design is essentially complete. The purpose of this review will be to (1) determine that the detail design of the configuration item under review satisfies the performance and engineering specialty requirements of the HWCI development specifications, (2) establish the detail design compatibility among the configuration item and other items of equipment, facilities, computer software and personnel, (3) assess configuration item risk areas (on a technical, cost, and schedule basis), (4) assess the results of the producibility analyses conducted on system hardware, and (5) review the preliminary hardware product specifications. For CSCIs, this review will focus on the determination of the acceptability of the detailed design performance, and test characteristics of the design solution, and on the adequacy of the operation and support documents."

Reference: DoD, MIL-STD-1521B, Military Standards, Technical Reviews and Audits for Systems, Equipments, and Computer Software, 4 June 1985.

Model Analysis Definition

Model Analysis. The semi-formal analysis of UML artifacts and supporting documentation that can range from qualitative assessment to a detailed quantitative study.

Carnegie Mellon
Software Engineering Institute

Remaining IDR SW Evaluations

IDR Software Evaluations 2

Areas of Concern	Suggested Detail/Action	Potential UML Artifact
CSCI Security (Information Assurance (IA))	• Identify unique IA requirements and techniques used for implementing and maintaining security within the CSCI	
CSCI Control Functions	• High level description of executive control and start/recovery features	Interaction overview diagram (activity diagram variant)
CSCI Resources Allocation	• Overall view of resources allocation across all CSCIs including timing, sequencing requirements and relevant constraints	Profile for Schedulability, Performance, and Time

Note: Blank cell means no applicable potential UML artifact. UML may be insufficient to document all aspects

IDR Software Evaluations 3

Areas of Concern	Suggested Detail/Action	Potential UML Artifact
Quality Requirements • Reliability • Maintainability • Supportability • Producibility • Safety • Extensibility • Flexibility • Reconfigurability • Interoperability	• Evaluate initial software designs against the quality requirements in the TRD. - Does the design meet these? If so, to what extent? - To what extent do they exceed or not meet the thresholds, the objectives? - Is there a plan to meet those missed? - Will the plan be executable in time for FDR, Milestone B? - Identify tradeoffs between the quality requirements? Are they acceptable? What risks are introduced? • Evaluations should be done from the prototype perspective identifying those components that are most likely to be evolved during Phase B.	Model Analysis

IDR Software Evaluations 4

Areas of Concern	Suggested Detail/Action	Potential UML Artifact
Human Factors including natural environment	• Evidence of functional integrity of the man with the machine to accomplish system operation • Ensure human performance requirements of TRD are met such as display content, control and data entry devices, error detection, outputs and formats. • Judge adequacy of human usability. • Ensure human limitations are not exceeded • Approach to climatic conditions • Adequate display of environment data	• Use cases • Sequence diagrams • Activity diagrams

IDR Software Evaluations 5

Areas of Concern	Suggested Detail/Action	Potential UML Artifact
Test	• Review test concepts including organization and responsibilities • Does methodology to be used meet the quality assurance requirements • Review interface test requirements • Review any existing test data • Review/inspect all test devices, mock-ups etc for program impact • Review software unit testing plans for - Address sizing, timing and accuracy - Present general and specific requirements being tested - Describe test unique support software, if any - Describe how the test unique software will be obtained - Provide test schedules that are consistent with higher level plans • Review CSC integration testing - Type of testing required - Requirements tested - Describe test unique support software and how obtained - Describe test management - Test schedules consistent with higher level test plans • Review CSCI testing plans - Objectives and relationship to other tests - Test environment including GFP - Test roles and responsibilities - Test schedules consistent with higher level plans - Requirements assigned to which test	• Deployment Diagram

IDR Software Evaluations 6

Areas of Concern	Suggested Detail/Action	Potential UML Artifact
Changes to Software Development Environment • CSCI Tools • Development Approach • Test • Personnel	Changes to baseline environment: • Describe availability, adequacy, and planned utilization of facilities. • Define and discuss any unique development features of the software that will not be in the operational software. • Provide details of Software Development Library. • Describe any development or test tools and/or software that will be used but not delivered under the contract • Describe any changes since initial environment created	

Note: Blank cell means no applicable potential UML artifact. UML may be insufficient to document all aspects

IDR Software Evaluations 7

Areas of Concern	Suggested Detail/Action	Potential UML Artifact
Sustainment • Support Resources • Software Test Equipment • Technical Manuals	• Describe resources needed to support software and firmware during Phase B and subsequent operational deployment such as personnel, special skills, human factors, configuration management, facilities/space. • Review software test equipment reliability/maintainability/availability • Review status - All agencies apprised - Suitability - Availability during DT&E - Review process	

Note: Blank cell means no applicable potential UML artifact. UML may be insufficient to document all aspects

Carnegie Mellon
Software Engineering Institute

Remaining FDR SW Evaluations

FDR Software Evaluations 2

Areas of Concern	Suggested Detail/Action	Potential UML Artifact
Database Design	• Detailed characteristics of databases including structure down to the item level • Rules for sharing, recovery, integrity, manipulation	• Class diagrams
Interface Design	• Detailed design characteristics of all interfaces including data source, destination, interface name and interrelationships • Overview of key design issues • Discuss format – fixed or subject to dynamic changes	• Sequence diagrams • State Charts

FDR Software Evaluations 3

Areas of Concern	Suggested Detail/Action	Potential UML Artifact
Quality Requirements	• Review quality attributes from architecture perspective - Reliability - Maintainability - Supportability - Producibility - Security - Safety - Extensibility - Flexibility - Reconfigurability - Interoperability • Reliability/maintainability/availability against TRD requirements and predictions of quantitative RMA • Review redundant CIs against IDR expectations • Review failure data reporting procedures and methods to determine trends • Review software reliability prediction model • Review safety design, operational maintenance safety analyses and procedures • Review acceptance test plan to ensure quality requirements are addressed	• Annotated class diagrams • Annotated sequence diagrams

FDR Software Evaluations 4

Areas of Concern	Suggested Detail/Action	Potential UML Artifact
Sustainment • Maintenance /Maintenance Data • Support Resources • Software Test Equipment • Technical Manuals	• Review unique maintenance procedures for CSCI during operational use including automatic, semi-automatic, and manual recovery from failures and malfunctions • Review software test equipment reliability/maintainability/availability • Review adequacy of maintenance plans • Review updates/progress since IDR	
Human Factors	• Review detailed design to ensure it meets human factors • Demonstrate adequacy of design for human performance • Review for man/machine compatibility • Evaluate: - Operator controls/displays - Maintenance / Safety features - Work space layout - Internal environmental conditions (noise, lighting, ventilation, etc) - Training equipment - Personnel accommodations	

Note: Blank cell means no applicable potential UML artifact. UML may be insufficient to document all aspects

FDR Software Evaluations 5

Areas of Concern	Suggested Detail/Action	Potential UML Artifact
Test	• Review test documentation for currency • Examine any test results against TRD hardware, software and interface requirements • Review quality assurance provisions • Inspect any breadboards, mockups or prototypes hardware for test program	• Use Case diagrams • Use Case specifications • Sequence diagrams