

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

Mary Ann Lapham

June 2006

Acquisition Support Program

Unlimited distribution subject to the copyright.

Technical Note
CMU/SEI-2006-TN-023

This work is sponsored by the U.S. Department of Defense.

The Software Engineering Institute is a federally funded research and development center sponsored by the U.S. Department of Defense.

Copyright 2006 Carnegie Mellon University.

NO WARRANTY

THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN "AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

Use of any trademarks in this report is not intended in any way to infringe on the rights of the trademark holder.

Internal use. Permission to reproduce this document and to prepare derivative works from this document for internal use is granted, provided the copyright and "No Warranty" statements are included with all reproductions and derivative works.

External use. Requests for permission to reproduce this document or prepare derivative works of this document for external and commercial use should be addressed to the SEI Licensing Agent.

This work was created in the performance of Federal Government Contract Number FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center. The Government of the United States has a royalty-free government-purpose license to use, duplicate, or disclose the work, in whole or in part and in any manner, and to have or permit others to do so, for government purposes pursuant to the copyright license under the clause at 252.227-7013.

For information about purchasing paper copies of SEI reports, please visit the publications portion of our Web site (<http://www.sei.cmu.edu/publications/pubweb.html>).

Contents

Abstract	vii
1 Introduction	1
2 IDR and FDR Definitions	3
3 Engineering Emphasis	7
4 IDR and FDR Criteria	9
4.1 Criteria Rationale.....	9
4.2 IDR and FDR Pre- and Postconditions.....	10
4.3 IDR and FDR Criteria Summaries	13
4.4 CSCI Preliminary Design Criteria Example	15
4.5 Applying the Criteria	16
5 Conclusion	18
Appendix A IDR Criteria	19
Appendix B FDR Criteria	23
Appendix C Acronyms	27
References	29

List of Figures

Figure 1: Notional DoD 5000 Acquisition Timeline	1
Figure 2: Example Incremental/Spiral Approach	6
Figure 3: Engineering Emphasis for Different Development Areas	8
Figure 4: IDR and FDR Validation Hierarchy	17

List of Tables

Table 1:	Characteristics of Development Approaches	5
Table 2:	IDR Preconditions	11
Table 3:	IDR Postconditions.....	12
Table 4:	FDR Preconditions	12
Table 5:	FDR Postconditions.....	13
Table 6:	Summary IDR Software Areas of Concern.....	14
Table 7:	Summary FDR Software Areas of Concern.....	15
Table 8:	Example IDR Software Evaluation Criteria.....	16

Abstract

Many Department of Defense (DoD) development programs, such as aircraft development programs, are typically complex and long-lived. Often, these programs are structured to demonstrate significant capability in the form of prototypes, which may be additionally intended to provide lingering operational capability. As such, technology development activities frequently include design reviews known as the Initial Design Review (IDR) and the Final Design Review (FDR) that are not present in most other systems acquisitions.

IDR and FDR content is not explicitly defined in regulations or policies; rather, it is defined by the program office. However, since IDR and FDR are the Technology Development phase's equivalent to Preliminary Design Review and Critical Design Review, this technical note proposes that they should have similar criteria, scaled for Technology Development work.

This technical note presents definitions of IDR and FDR, their context in the acquisition life cycle, a comparison of engineering emphasis during IDR and FDR, IDR and FDR pre- and postconditions, and IDR and FDR criteria and how to apply it. The audiences for this technical note are managers and developers of medium to large DoD systems that employ technology that is not mature enough to transition directly to systems development.

1 Introduction

Many DoD development programs, such as aircraft development programs, are typically complex and long-lived. Often, these programs are structured to demonstrate significant capability in the form of prototypes prior to Milestone B in the Defense Acquisition Management Framework [DoD 03]. This early technology work accelerates and facilitates the application of mature, advanced technologies to DoD programs. As such, these programs frequently include design reviews that are not present in most other systems acquisition life cycles. These supplemental design reviews are most frequently known as the Initial Design Review (IDR) and the Final Design Review (FDR). The IDR and FDR occur during the Technology Development phase of the Defense Acquisition Management Framework [DoD 03]

Following the aircraft development program example, the timeline shown in Figure 1 denotes a notional aircraft acquisition schedule as it relates to the DoD acquisition life cycle. The figure shows the relationship of the IDR and FDR to the overall milestones and other reviews such as Preliminary Design Review (PDR) and Critical Design Review (CDR). As shown, the IDR and FDR are in the Technology Development (TD) phase of the program. The TD phase is the time frame between Milestone A and Milestone B. The PDR, CDR, and Test Readiness Review (TRR) are in the System Development and Demonstration (SDD) phase of the life cycle. The SDD phase occurs between Milestone B and Milestone C.

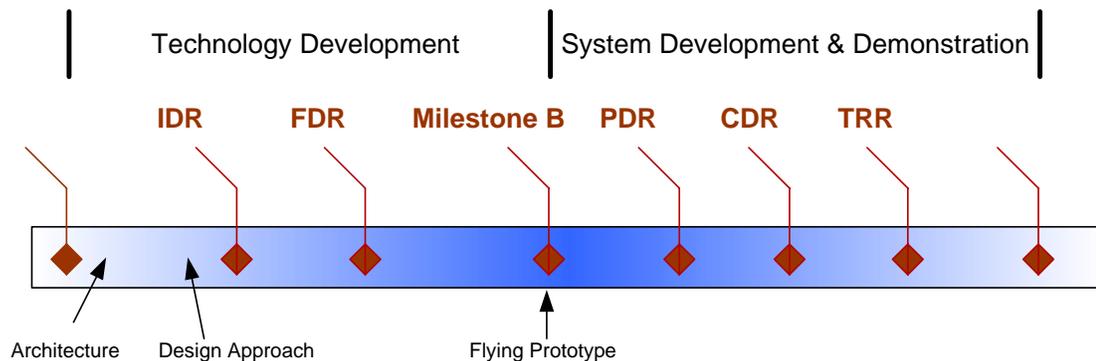


Figure 1: Notional DoD 5000 Acquisition Timeline

At the end of the TD phase, it is anticipated that working aircraft prototypes exist for evaluation. These prototypes most likely include all segments of the program (radar, platform, munitions, etc.). Given that the prototypes contain a good portion of the capabilities for the aircraft, the IDR and FDR should ensure the required capability is well represented during the demonstration of the prototype and that high-risk items have been addressed to mitigate the risk as much as possible at this time. In addition, the IDR and FDR

checkpoints help fulfill the need for increasingly rigorous architectural reviews of system-level capabilities to verify desired emergent properties of the system such as

- security
- flexibility
- extensibility
- configurability
- interoperability
- and others, as identified for the particular system

However, the IDR and FDR content is not explicitly defined in any regulation or policy; rather, they are defined by the program office. Thus, there is a dilemma for the program office. With no guidelines to follow, the program office needs to grapple with what should be included in the IDR and FDR criteria. The remainder of this technical note provides a definition for IDR and FDR, some generic criteria, and some suggestions on how to apply the criteria.

The audiences for this technical note are managers and developers of medium to large DoD systems that employ technologies that are not mature enough to transition directly to systems development.

2 IDR and FDR Definitions

No specific regulation or policy defines an IDR or FDR; rather, the guidance states that these reviews are defined by the program office. IDR and FDR do fulfill a roughly equivalent purpose in the TD phase as the PDR and CDR do in the SDD phase by providing a way to

- evaluate developed system and software architecture and design
- ensure that desired and required capabilities are well represented during prototype demonstration
- ensure high-risk items have been addressed to mitigate the risk as much as possible
- document engineering and management decisions

Given this purpose, the following are working definitions for IDR and FDR:

IDR – formal technical review of prototype design approach for a configuration item (CI) or computer software configuration item (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.

Note that the only difference between these definitions is the addition of one word in the FDR definition—*detailed*. In this instance, “detailed” means that the design contains sufficient information that it could be handed over to a separate development team that could successfully complete the work. While these definitions apply to both hardware and software components of a system, this technical note only discusses CSCIs.

PDR and CDR content is explicitly defined in MIL-STD-1521B. While MIL-STD-1521B, *Military Standard Technical Reviews and Audits for Systems, Equipments, and Computer Software*, was cancelled April 10, 1995, it is still used today to provide guidance for PDR and CDR technical reviews [DoD 85]. The functional configuration audit and physical configuration audit requirements originally included in MIL-STD-1521B were incorporated into MIL-STD-973, *Configuration Management* [DoD 92], but all other areas of MIL-STD-1521B were cancelled and not replaced. MIL-STD-973 was cancelled September 30, 2000.

As mentioned earlier, since the IDR and FDR fulfill a roughly equivalent purpose for the TD phase as the PDR and CDR do for the SDD phase, MIL-STD-1521B could also form the basis for IDR and FDR criteria. Of course, the criteria would need to be tailored to meet the needs of the specific program’s TD phase.

The IDR and FDR criteria described in this technical note are based on MIL-STD-1521B. In addition, the U.S. Air Force's *Software Management Guidebook* provides some additional guidance that was used to create the IDR and FDR criteria (in particular, the entrance and exit criteria) presented here [USAF 04].

A few words of caution are in order. MIL-STD-1521B assumed a waterfall approach to software development and functional decomposition of requirements. Today's development environments often incorporate an incremental or spiral development approach, which can affect how the IDR and FDR criteria are applied.

The waterfall approach consists of the requirements development, design, build, unit test, system integration and test, and field steps. In the waterfall method, one step is completed before continuing to the next step. Attributes of the waterfall approach include well-defined, low volatility requirements, a single development cycle, a small and precedented system, and no distribution of interim software. Typically, the hardware and software both follow this cycle, merging at the system integration and test phase.

In the incremental development approach, there can be several cycles of the design, build, and test steps and there is feedback from one cycle to the next. Some of the increments may or may not be fielded. The hardware and software cycles can use different development approaches (i.e., hardware could use the waterfall approach with software using incremental). The attributes of the incremental approach include well-defined, low to moderate volatility requirements, multiple development cycles, any size and possibly unprecedented system, and distribution of interim software.

Finally, the spiral approach is a risk-driven development model. The spiral approach uses cyclic concurrent engineering for which the process and product are determined by risk. This approach grows a system via risk-driven experimentation and elaboration, thus lowering development cost by early elimination of nonviable alternatives and avoiding rework. The attributes of the spiral approach include not well-defined, moderate to high volatility requirements, multiple development cycles, any size and possibly unprecedented system, and distribution of interim software.

Table 1 summarizes the characteristics of each development approach.

Table 1: Characteristics of Development Approaches

Development Approach \ Characteristics	Waterfall	Incremental	Spiral
Quality of Requirements	Good	Good	Poor
Requirements Volatility	Low	Low to Moderate	Moderate to High
Number of Development Cycles	Single	Multiple	Multiple
System Size	Small	Any	Any
System Precedence	Has Precedence	Possibly Unprecedented	Unprecedented
Interim Software Releases?	No	Yes	Yes

With the incremental or spiral approach, a complete functional decomposition of the requirements does not necessarily occur at the beginning of the program; rather, it occurs as the increments are defined or as the program evolves during the spirals. When using the incremental or spiral development approach, there will be multiple IDRs and FDRs, a set occurring for each increment or spiral. Figure 2 shows an example of how the incremental and spiral approaches can be combined during development of a system and where the IDRs and FDRs fit in. A combination of approaches may be used when most of the increments are well defined, but for example, a spiral approach is needed for a particular portion of the system due to risk, lack of technology maturity, or volatile requirements. When using the IDR and FDR criteria proposed in this technical note, keep in mind that the program must take the development environment into account.

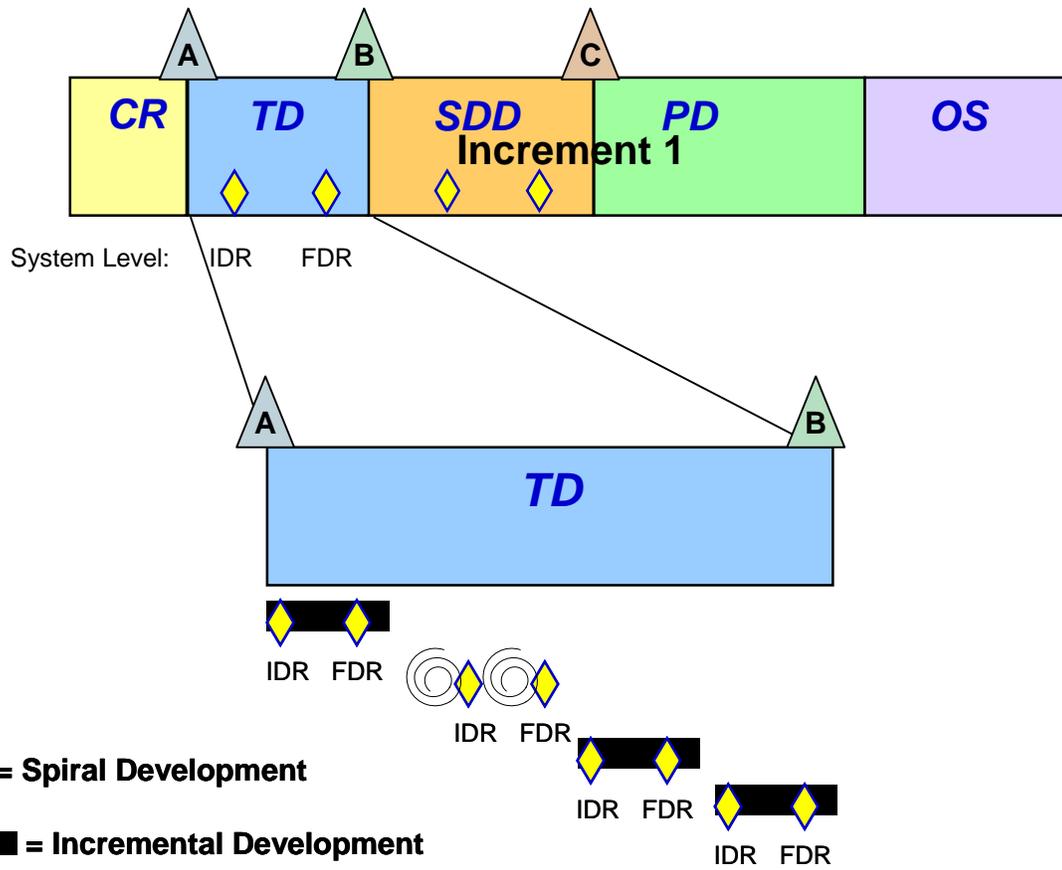


Figure 2: Example Incremental/Spiral Approach

3 Engineering Emphasis

During our review of MIL-STD-1521B and the U.S. Air Force's *Software Management Guidebook*, we identified 12 software development areas that form the basis of the IDR and FDR content:

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)

In addition to identifying the above development areas, we estimated the level of engineering effort or emphasis that would be expended during the IDR and FDR. The estimation process was based on a small survey of experienced software personnel. Each person was asked to rate the amount of emphasis the specific development area should receive during IDR and FDR, using the scale "low," "medium," and "high." We then formed a general consensus to provide a relative idea of how much work should be expended in each area during IDR and FDR. These results can be used as a rule of thumb and as high-level guidance for planning purposes. For instance, one would expect a high level of emphasis for preliminary design during IDR and a low level of emphasis on preliminary design during the FDR. In fact, the work related to preliminary design during FDR would be "cleanup" in nature, such as clarifying any outstanding issues leftover from the IDR milestone.

Figure 3 shows the comparison of IDR and FDR engineering emphasis for each of the 12 areas we identified.

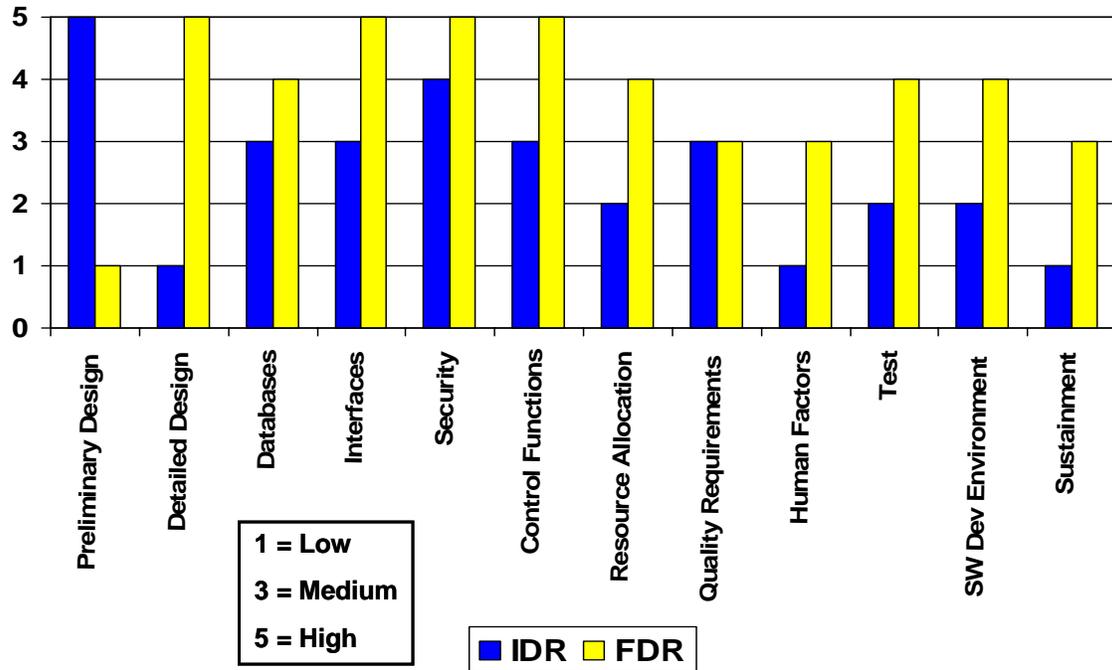


Figure 3: Engineering Emphasis for Different Development Areas

4 IDR and FDR Criteria

The IDR and FDR are technically milestones, but the processes leading up to the milestones include a wide range of activities that must be accomplished before the actual milestone can be deemed a success. Therefore, the IDR and FDR milestones are capstone milestones for which a review is done on the engineering activities that preceded them. Thus, there must be criteria created for each milestone. In addition, preconditions (entry) and postconditions (exit) criteria for both IDR and FDR must be defined. The rationale for the criteria, the pre- and postconditions, and the actual criteria are described in this section.

4.1 Criteria Rationale

Many of today's large defense programs use Advanced Concept Technology Demonstrations (ACTDs) to gain understanding and evaluate the utility of a technology, develop a concept of operations for that technology, and expedite delivery of new capabilities to combat forces. ACTDs promote the rapid transition of the new technology into the appropriate phase of a formal acquisition program. Programs operating in an ACTD-like environment develop initial capabilities in prototype form. These prototypes, rather than merely acting as a proof of concept or refinement of requirements, are intended to provide lingering operational capability for extended operational evaluation and exploitation.

A system and software architecture and design must be developed in the TD phase of the acquisition life cycle and evaluated at IDR and FDR. The system representation must include a definition of the configuration items (CIs) for both hardware and software, how functionality is distributed across CIs, and how they interact to provide system-level capabilities.

These system-level capabilities include a number of necessary attributes that are architectural in nature. These attributes include but are not limited to security, flexibility, extensibility and reconfigurability, and interoperability. Evaluation of these architectural attributes must be supported by increasingly rigorous architectural reviews to verify that the desired emergent properties of the system will, indeed, be present when the system is fielded.

In addition to the emergent system properties, a number of workload, resource, and logistics attributes that are more characteristic of the design than the architecture need to be verified. This is because the TD phase is intended to provide operational utility that needs to be supported and evolved in the development phase. The architectural and design reviews can be combined for IDR and FDR.

Another important aspect of IDR and FDR criteria falls in the engineering and management area. As the program progresses, the comprehensive design and decision rationale should be recorded. Many times, this is not achieved. This is especially true when a program employs incremental or spiral-based development approaches. Often there are a collection of “hard” requirements that induce disproportionate risk on the system. It has been observed in both an incremental and spiral development of capabilities for demonstration, that there is a tendency to defer the “difficult requirements” to a later increment or spiral in order to produce partial capability for demonstration. Strong IDR and FDR criteria are needed to ensure that program development progresses in such a way that hard capabilities, which represent risks to the program, are associated with effective mitigation and development activities earlier in the project life cycle. In addition, these mitigation and development activities need to be well documented so that future program participants can track why and perhaps how certain decisions were made.

4.2 IDR and FDR Pre- and Postconditions

For both IDR and FDR, a set of conditions is defined for entry and exit into the IDR and FDR milestones. These criteria are demonstrable conditions in that they require something to be created, available, or accomplished before the IDR or FDR milestone can be achieved.

The proposed sets of IDR and FDR pre- and postconditions are found in Table 2 through Table 5. Each table contains two columns. The first column is labeled either “Preconditions” or “Postconditions.” These are the lists of items that need to be created, available, or accomplished for IDR or FDR to either enter into the review or exit the review. The second column in each table is labeled “Potential UML Artifacts.” This column provides some insight into common Unified Modeling Language (UML) artifacts that may be helpful in satisfying the IDR and FDR artifact requirements. Note that “Potential UML Artifacts” column may contain blank cells. This indicates that no applicable UML artifact exists to satisfy that condition. Although UML is the modeling and specification language of choice for many development efforts, it may be insufficient to document all aspects of your software development. In this case, other documentation such as specifications, interface control documents, and design documents should be used. In some instances, a “+Model analysis” entry is listed in the “Potential UML Artifacts” column. This indicates that an actual analysis of the model may provide additional information to meet the cited conditions.

Table 2: IDR Preconditions

Preconditions	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
Make / buy decisions (legacy and commercial off-the-shelf [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	

Table 3: IDR Postconditions

Postconditions	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: <ul style="list-style-type: none"> • Interface Control Document • Software metrics • Software Test Plan • COTS and reusable software • Software development process • Software estimates 	
Life-cycle software support requirements update	
Software item approved to start detailed design	

Table 4: FDR Preconditions

Preconditions	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

Table 5: FDR Postconditions

Postconditions	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

4.3 IDR and FDR Criteria Summaries

In Section 3, Engineering Emphasis, we identified twelve areas of software development that form the basis of the IDR and FDR evaluation criteria. Note that not all twelve areas are represented in the pre- and postcondition criteria for IDR and FDR. For instance, “Detailed Design” is not discussed in Table 2 or Table 3, nor is “Preliminary Design” discussed in Table 4 or Table 5.

The first column in Table 6 summarizes the areas of concern addressed for IDR; the first column in Table 7 summarizes the areas of concern addressed in FDR. In each table, the second column lists potential UML artifacts. As with Table 2 through Table 5, blank cells in the column labeled “Potential UML Artifact” indicate that an applicable UML artifact for that area of concern does not exist.

Table 6: Summary IDR Software Areas of Concern

Areas of Concern	Potential UML Artifacts
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 IA (Information Assurance)	
CSCI Control Functions	Interaction overview diagram (activity diagram variant)
CSCI Resources Allocation	Profile for schedulability, performance, and time
Quality Requirements: <ul style="list-style-type: none"> • Reliability • Maintainability • Supportability • Producibility • Safety • Extensibility • Flexibility • Reconfigurability • Interoperability 	+ Model analysis
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	

Table 7: Summary FDR Software Areas of Concern

Areas of Concern	Potential UML Artifacts
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

4.4 CSCI Preliminary Design Criteria Example

Table 8 shows the basic criteria we identified for the CSCI Preliminary Design software development area, including the suggested detail/action and the potential UML artifacts for IDR. In this case, the area of concern includes functional flow, CSCI structure, interfaces, and COTS / GOTS / Reuse. A suggested detail is a top-level description of the functional flow for all software technical requirements document (TRD) requirements, including interfaces. Example UML artifacts are sequence or swim lane diagrams.

Table 8: Example IDR Software Evaluation Criteria

Areas of Concern	Suggested Details / Actions	Potential UML Artifacts
CSCI Preliminary Design Functional Flow CSCI Structure Interfaces COTS / GOTS / Reuse	<ul style="list-style-type: none"> • 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)

Due to the extensive nature of the IDR and FDR criteria, comprehensive lists of specific criteria are included in Appendices A and B, respectively.

4.5 Applying the Criteria

IDR and FDR are the culmination of the activities that generated the information required to exit each event. These formal, technical reviews are the result of significant work done at the working group level and in various intermediate reviews. depicts how the criteria can be applied at different levels of abstraction during a program. The bottom layer shows the reviews supported by the working group, integrated product team (IPT) or technical interchange meeting (TIM). The middle layer, or “wall walk,” is the intermediate review at the CI or CSCI level, which is conducted in preparation for the formal IDR or FDR. The top of the pyramid represents the capstone IDR or FDR event.

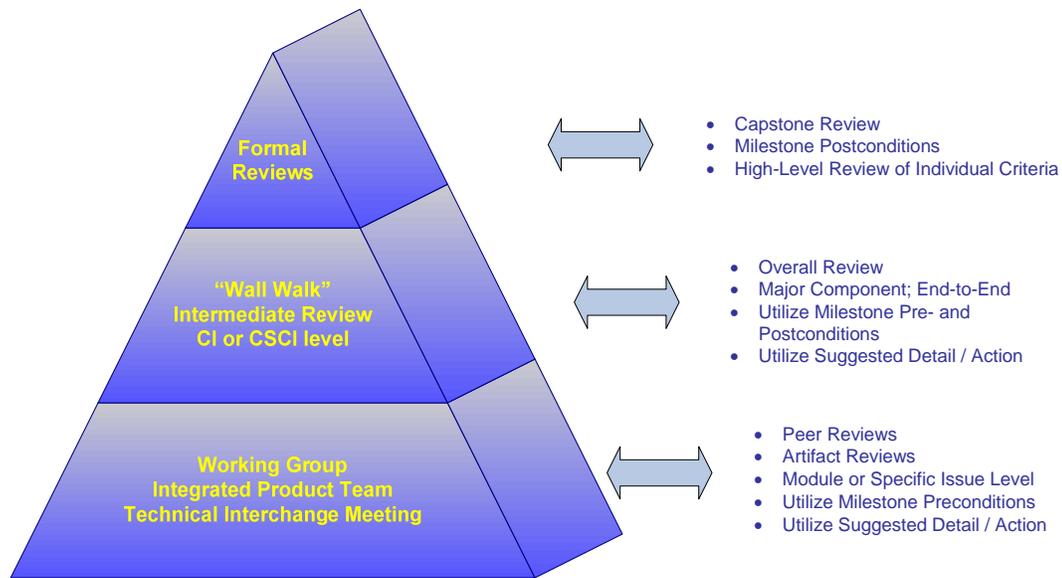


Figure 4: IDR and FDR Validation Hierarchy

At the lowest level (the working group, IPT, and TIM level), peer reviews and artifact reviews are conducted by applying the suggested IDR or FDR criteria listed in the tables in Appendices A or B, respectively. All milestone preconditions must be met in order to start this phase. During this review, module or specific issues are raised, researched, resolved, and/or mitigated. After work is completed at this level, the reviews can proceed to the intermediate level or “wall walk” level.

The intermediate level is sometimes called “wall walk” because the data may be presented for review tacked to the walls of a large room. Reviewers walk around or along the wall and examine the material. The preconditions must be satisfied to enter this phase and the reviewers should ensure that the postconditions can be met by evaluating the data presented. This review is an overall, end-to-end review of major components. The major components usually comprise multiple components or modules reviewed at the working group level. Again, the suggested IDR or FDR criteria listed in the tables in Appendices A or B, respectively, is applied to the higher level components.

Finally, the capstone event, the IDR or FDR, is held. This is a high-level review of individual criteria that provides the evidence necessary for the milestone postconditions to be declared met. All the stakeholders are invited to the IDR and FDR. The Program Office has final say, with inputs from the stakeholders, as to whether the postconditions are met. These reviews vary in length, depending on the complexity of the program and how much of the program is covered in the review. In many instances, a large program is divided into segments with each segment holding its own IDR, FDR, and so on.

5 Conclusion

IDRs and FDRs held during the TD phase of a program can ensure that the required capability is demonstrated and that high risks have been mitigated. They provide a way to introduce rigor and formality into a program during the TD phase, ensuring a required level of progress, technical adequacy, and risk resolution on a technical, cost, and schedule basis has been achieved.

Readers of this technical note are encouraged to adapt the IDR and FDR criteria presented to their program's environment and needs. Users must also decide what non-UML artifacts are needed to round out the program's software documentation package. These adapted criteria can then be used to provide guidance to the contractors in the software area. In addition, the capstone milestones of IDR and FDR can be added to the Integrated Management Plan and Integrated Master Schedule of the program. This will ensure that the events are well known and that the activities necessary to achieve successful IDR and FDR are planned and accomplished.

Appendix A IDR Criteria

IDR Software Evaluation Criteria

Areas of Concern	Suggested Details / Actions	Potential UML Artifacts
CSCI Preliminary Design Functional Flow CSCI Structure Interfaces COTS / GOTS / Reuse	<ul style="list-style-type: none"> • 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)
CSCI Security IA (Information Assurance)	<ul style="list-style-type: none"> • Identify unique IA requirements and techniques used for implementing and maintaining security within the CSCI. 	
CSCI Control Functions	<ul style="list-style-type: none"> • High-level description of executive control and start/recovery features 	Interaction overview diagram (activity diagram variant)
CSCI Resources Allocation	<ul style="list-style-type: none"> • Overall view of resources allocation across all CSCIs including timing, sequencing requirements, and relevant constraints 	Profile for schedulability, performance, and time

Areas of Concern	Suggested Details / Actions	Potential UML Artifacts
Quality Requirements <ul style="list-style-type: none"> • Reliability • Maintainability • Supportability • Producibility • Safety • Extensibility • Flexibility • Reconfigurability • Interoperability 	<ul style="list-style-type: none"> • 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
Human Factors including natural environment	<ul style="list-style-type: none"> • 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
Changes to Software Development Environment CSCI Tools Development Approach Test Personnel	Changes to baseline environment: <ul style="list-style-type: none"> • 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. 	

Areas of Concern	Suggested Details / Actions	Potential UML Artifacts
Sustainment Support Resources Software Test Equipment Technical Manuals	<ul style="list-style-type: none"> • Describe resources needed to support software and firmware during Phase B and subsequent operational deployment such as personnel, special skills, human factors, configuration management, and facilities/space. • Review software test equipment reliability/maintainability/availability. • Review status. • All agencies apprised • Suitability • Availability during Developmental Test and Evaluation (DT&E) • Review process 	

Appendix B FDR Criteria

FDR Software Evaluation Criteria

Areas of Concern	Suggested Details / Actions	Potential UML Artifacts
Software Detailed Design	<ul style="list-style-type: none"> • Establish integrity of software design. • Identify additional in-progress reviews as needed. • Action items • Engineering Change Proposal (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 Specification. • 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 	<p>Class diagrams</p> <p>Sequence diagrams</p> <p>State charts</p>
Database Design	<ul style="list-style-type: none"> • Detailed characteristics of databases including structure down to the item level. • Rules for sharing, recovery, integrity, manipulation 	<p>Class diagrams</p>

Areas of Concern	Suggested Details / Actions	Potential UML Artifacts
Interface Design	<ul style="list-style-type: none"> • 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 	<p>Sequence diagrams</p> <p>State charts</p>
Quality Requirements	<ul style="list-style-type: none"> • Review quality attributes from architecture perspective: <ul style="list-style-type: none"> ○ Reliability ○ Maintainability ○ Supportability ○ Producibility ○ Security ○ Safety ○ Extensibility ○ Flexibility ○ Reconfigurability ○ Interoperability • Reliability/maintainability/availability (RMA) 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. 	<p>Annotated class diagrams</p> <p>Annotated sequence diagrams</p>
Sustainment Maintenance / Maintenance Data Support Resources Software Test Equipment Technical Manuals	<ul style="list-style-type: none"> • 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. 	

Areas of Concern	Suggested Details / Actions	Potential UML Artifacts
Human Factors	<ul style="list-style-type: none"> • Review detailed design to ensure it meets human factors • Demonstrate adequacy of design for human performance • Review for man/machine compatibility • Evaluate: <ul style="list-style-type: none"> ○ Operator controls and displays ○ Maintenance / Safety features ○ Work space layout ○ Internal environmental conditions (noise, lighting, ventilation, etc) ○ Training equipment ○ Personnel accommodations 	
Test	<ul style="list-style-type: none"> • 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. 	<p>Use case diagrams</p> <p>Use case specifications</p> <p>Sequence diagrams</p>

Appendix C Acronyms

ACTD	Advanced Concept Technology Demonstration
CDR	Critical Design Review
CI	Configuration Item
COTS	Commercial Off-the-Shelf
CSCI	Computer Software Configuration Item
DoD	Department of Defense
DT&E	Developmental Test and Evaluation
ECP	Engineering Change Proposal
FDR	Final Design Review
GFP	Government Furnished Property
GOTS	Government Off-the-Shelf
HW	Hardware
IA	Information Assurance
IDR	Initial Design Review
IPT	Integrated Product Team
PDR	Preliminary Design Review
RMA	Reliability/Maintainability/Availability
SDD	System Development and Demonstration
SW	Software
TD	Technology Development
TIM	Technical Interchange Meeting

TRD Technical Requirements Document

TRR Test Readiness Review

USAF United States Air Force

References

URLs are valid as of the publication date of this document.

- [DoD 85] Department of Defense. *MIL-STD-1521B (USAF) Military Standard Technical Reviews and Audits for Systems, Equipments, and Computer Software*. (Cancelled April 10, 1995)
http://ax.losangeles.af.mil/se_revitalization/AA_Functions/Design_Review/Attachments/Mil-Std_1521B.PDF (1985).
- [DoD 92] Department of Defense. *MIL-STD-973 Military Standard Configuration Management*. (Cancelled September 30, 2000)
<http://www.dscr.dla.mil/qap/milstd973.pdf> (1992).
- [DoD 03] Department of Defense. *DoD Instruction Operation of the Defense Acquisition System (DoDI 5000.2)*.
<http://www.dtic.mil/whs/directives/corres/pdf2/i50002p.pdf> (2003).
- [USAF 04] U.S. Air Force. *Software Management Guidebook, Version 0.9*.
<https://ossg.gunter.af.mil/applications/sep/documents/bk\af software management guidebook.doc> (2004).

REPORT DOCUMENTATION PAGE			<i>Form Approved</i> <i>OMB No. 0704-0188</i>	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.				
1. AGENCY USE ONLY (Leave Blank)	2. REPORT DATE June 2006	3. REPORT TYPE AND DATES COVERED Final		
4. TITLE AND SUBTITLE Specifying Initial Design Review (IDR) and Final Design Review (FDR) Criteria		5. FUNDING NUMBERS FA8721-05-C-0003		
6. AUTHOR(S) Mary Ann Lapham				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Software Engineering Institute Carnegie Mellon University Pittsburgh, PA 15213		8. PERFORMING ORGANIZATION REPORT NUMBER CMU/SEI-2006-TN-023		
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) HQ ESC/XPK 5 Eglin Street Hanscom AFB, MA 01731-2116		10. SPONSORING/MONITORING AGENCY REPORT NUMBER		
11. SUPPLEMENTARY NOTES				
12A DISTRIBUTION/AVAILABILITY STATEMENT Unclassified/Unlimited, DTIC, NTIS		12B DISTRIBUTION CODE		
13. ABSTRACT (MAXIMUM 200 WORDS) Many Department of Defense (DoD) development programs, such as aircraft development programs, are typically complex and long-lived. Often, these programs are structured to demonstrate significant capability in the form of prototypes, which may be additionally intended to provide lingering operational capability. As such, technology development activities frequently include design reviews known as the Initial Design Review (IDR) and the Final Design Review (FDR) that are not present in most other systems acquisitions. IDR and FDR content is not explicitly defined in regulations or policies; rather, it is defined by the program office. However, since IDR and FDR are the Technology Development phase's equivalent to Preliminary Design Review and Critical Design Review, this technical note proposes that they should have similar criteria, scaled for Technology Development work. This technical note presents definitions of IDR and FDR, their context in the acquisition life cycle, a comparison of engineering emphasis during IDR and FDR, IDR and FDR pre- and postconditions, and IDR and FDR criteria and how to apply it. The audiences for this technical note are managers and developers of medium to large DoD systems that employ technology that is not mature enough to transition directly to systems development.				
14. SUBJECT TERMS acquisition, commercial off-the-shelf, COTS, database, documentation, integration, operation, sustainment, software architecture, security, spiral development, test, training, risk, risk management		15. NUMBER OF PAGES 40		
16. PRICE CODE				
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL	