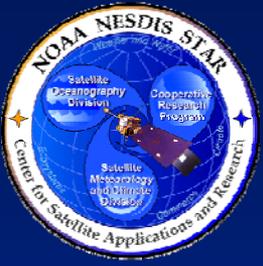




NOAA NESDIS CENTER for SATELLITE APPLICATIONS and RESEARCH

DOCUMENT GUIDELINE DG-8.2.A

CRITICAL DESIGN DOCUMENT GUIDELINE - APPENDIX Version 3.0



**TITLE: DG-8.2.A: CRITICAL DESIGN DOCUMENT
GUIDELINE APPENDIX VERSION 3.0**

AUTHORS:

Ken Jensen (Raytheon Information Solutions)

DATE: October 1, 2009



<Project and/or Product Name>

Critical Design Review

<Date>

Prepared By: <Preparer 1>¹, <Preparer 2>², ..., and <Preparer N>^N

¹ <Organization for Preparer 1>

² <Organization for Preparer 2> ...

^N <Organization for Preparer N>



Review Agenda

Introduction	<Start Time> - <End Time>	<Presenter(s)>
PDR Report	<Start Time> - <End Time>	<Presenter(s)>
Operations Concept	<Start Time> - <End Time>	<Presenter(s)>
Requirements	<Start Time> - <End Time>	<Presenter(s)>
Break	<Start Time> - <End Time>	
Algorithm Theoretical Basis	<Start Time> - <End Time>	<Presenter(s)>
Software Architecture & Interfaces	<Start Time> - <End Time>	<Presenter(s)>
Lunch	<Start Time> - <End Time>	
Detailed Design Description	<Start Time> - <End Time>	<Presenter(s)>
Quality Assurance	<Start Time> - <End Time>	<Presenter(s)>
Break	<Start Time> - <End Time>	
Requirements Allocation	<Start Time> - <End Time>	<Presenter(s)>
Risks and Actions	<Start Time> - <End Time>	<Presenter(s)>
Summary and Conclusions	<Start Time> - <End Time>	<Presenter(s)>

Review Agenda Slide



Review Agenda - Day 1

Introduction	<Start Time> - <End Time>	<Presenter(s)>
PDR Report/Actions	<Start Time> - <End Time>	<Presenter(s)>
Break	<Start Time> - <End Time>	
Operations Concept	<Start Time> - <End Time>	<Presenter(s)>
Requirements	<Start Time> - <End Time>	<Presenter(s)>
Lunch	<Start Time> - <End Time>	
Algorithm Theoretical Basis	<Start Time> - <End Time>	<Presenter(s)>
Break	<Start Time> - <End Time>	
Software Architecture & Interfaces	<Start Time> - <End Time>	<Presenter(s)>

**Review Agenda Slide
Alternative – Day 1**



Review Agenda - Day 2

Detailed Design Description	<Start Time> - <End Time>	<Presenter(s)>
Break	<Start Time> - <End Time>	
Requirements Allocation	<Start Time> - <End Time>	<Presenter(s)>
Lunch	<Start Time> - <End Time>	
Quality Assurance	<Start Time> - <End Time>	<Presenter(s)>
Break	<Start Time> - <End Time>	
Risks and Actions	<Start Time> - <End Time>	<Presenter(s)>
Summary and Conclusions	<Start Time> - <End Time>	<Presenter(s)>

**Review Agenda Slide
Alternative – Day 2**



- **INTRODUCTION**
- PDR REPORT
- OPERATIONS CONCEPT
- REQUIREMENTS
- ALGORITHM THEORETICAL BASIS
- SOFTWARE ARCHITECTURE AND INTERFACES
- DETAILED DESIGN DESCRIPTION
- QUALITY ASSURANCE
- REQUIREMENTS ALLOCATION
- RISKS AND ACTIONS
- SUMMARY AND CONCLUSIONS



Section 1 - Introduction

Presented by

<Presenter's Name>
<Presenter's Title/Role>
<Presenter's Organization>



<Project Name> - Development Project Plan

- The Development Project Plan (DPP) is a standard artifact of the STAR EPL process.
 - » The DPP identifies project objectives, stakeholder roles and tasks, resources, milestones and schedule
 - » CDR reviewers can access this document at <pointer to the DPP>
- Guidelines for the DPP are found in STAR EPL process asset DG-5.1
 - » CDR reviewers can access this document at <pointer to DG-5.1>



Project Objectives

- **Objective 1**
 - » Sub-bullet 1
 - »
 - » Sub-bullet N
- **Objective 2**
 - » Sub-bullet 1
 - »
 - » Sub-bullet N
-
- **Objective M**
 - » Sub-bullet 1
 - »
 - » Sub-bullet N



Project Stakeholders

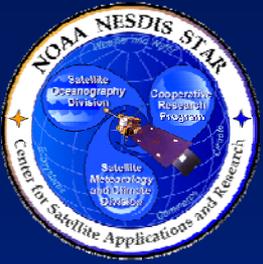
- **<Stakeholder Role 1> - <Named Stakeholder(s) or TBD>**
 - » Sub-bullet 1 (Description of stakeholder tasks)
 - »
 - » Sub-bullet M (Description of stakeholder tasks)
- **<Stakeholder Role 2> - <Named Stakeholder(s) or TBD>**
 - » Sub-bullet 1 (Description of stakeholder tasks)
 - »
 - » Sub-bullet M (Description of stakeholder tasks)
-
- **<Stakeholder Role N> - <Named Stakeholder(s) or TBD>**
 - » Sub-bullet 1 (Description of stakeholder tasks)
 - »
 - » Sub-bullet M (Description of stakeholder tasks)



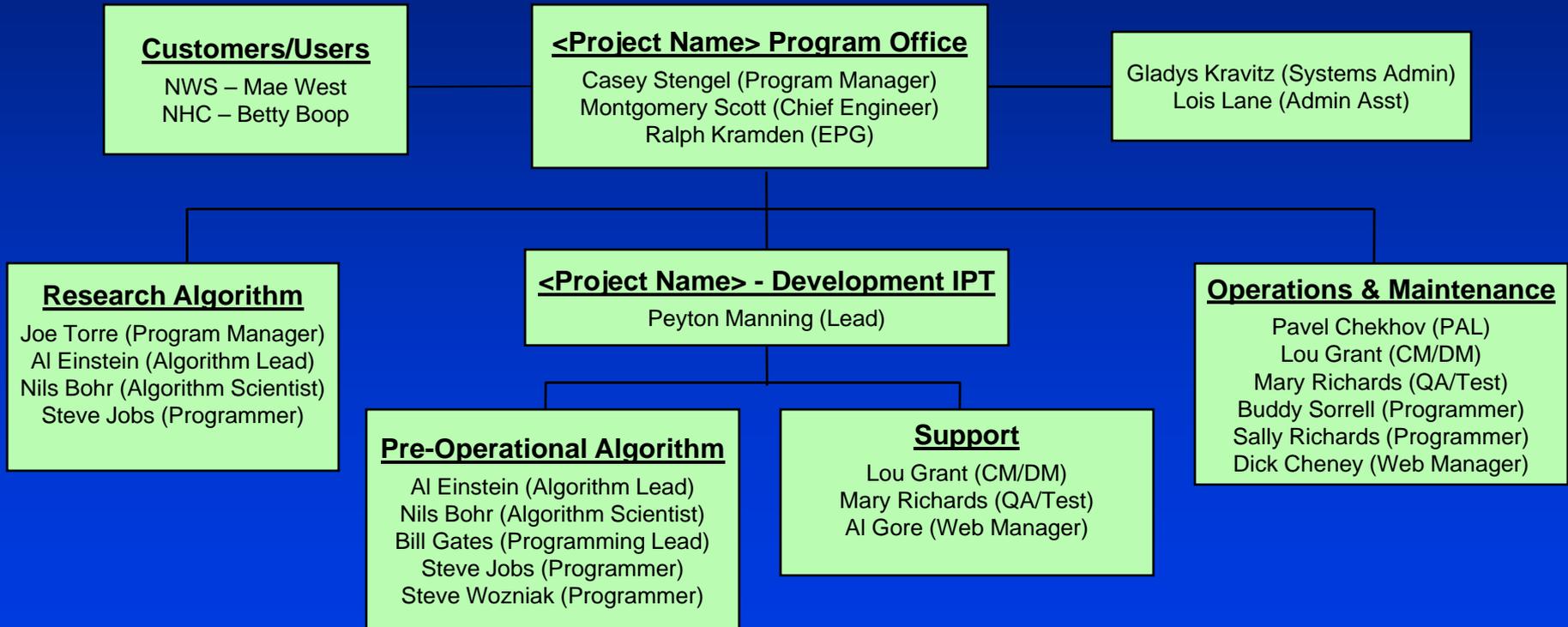
Project Stakeholders

Stakeholder	Names	Description
<Role 1>	<Names or TBD>	<Description>
<Role 2>	<Names or TBD>	<Description>
.....	<Names or TBD>	<Description>
<Role N>	<Names or TBD>	<Description>

**Section 1.3 –
Table Alternative**



<Project Name> Organization Chart





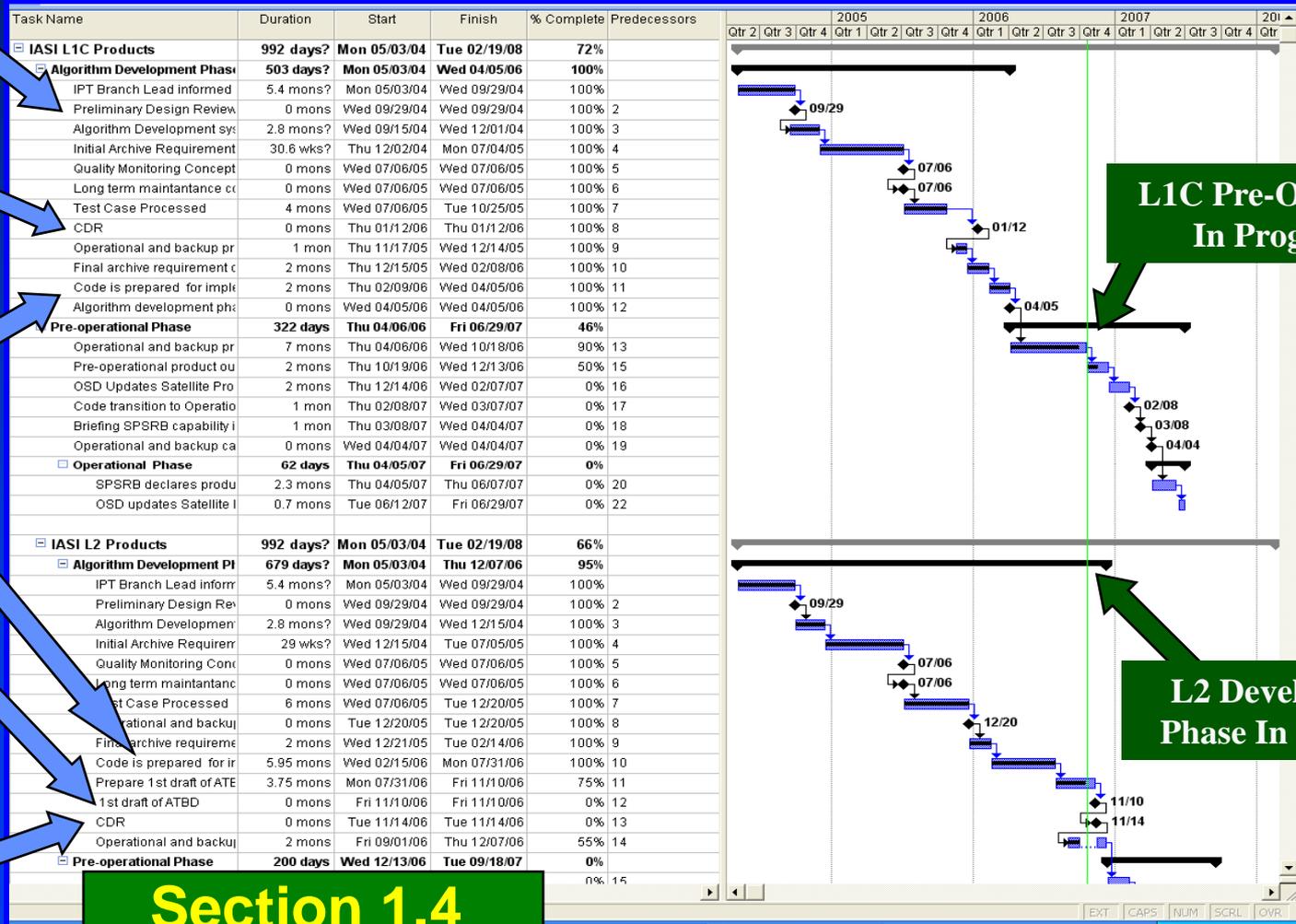
Project Milestones

- Gate 3 Review - <Date>
- Project Requirements Review - <Date>
- Preliminary Design Review - <Date>
- **Critical Design Review - <Date>**
- Gate 4 Review - <Date>
- Test Readiness Review - <Date>
- Code Test Review - <Date>
- System Readiness Review - <Date>
- Gate 5 Review - <Date>
- Delivery to Operations - <Date>



Project Timeline

- PDR**
09/29/04
- L1C Products CDR**
01/12/06
- L1C Code**
04/05/06
- L2 Code**
07/31/06
- ATBD 1st draft**
11/10/06
- L2 Products CDR**
11/14/06

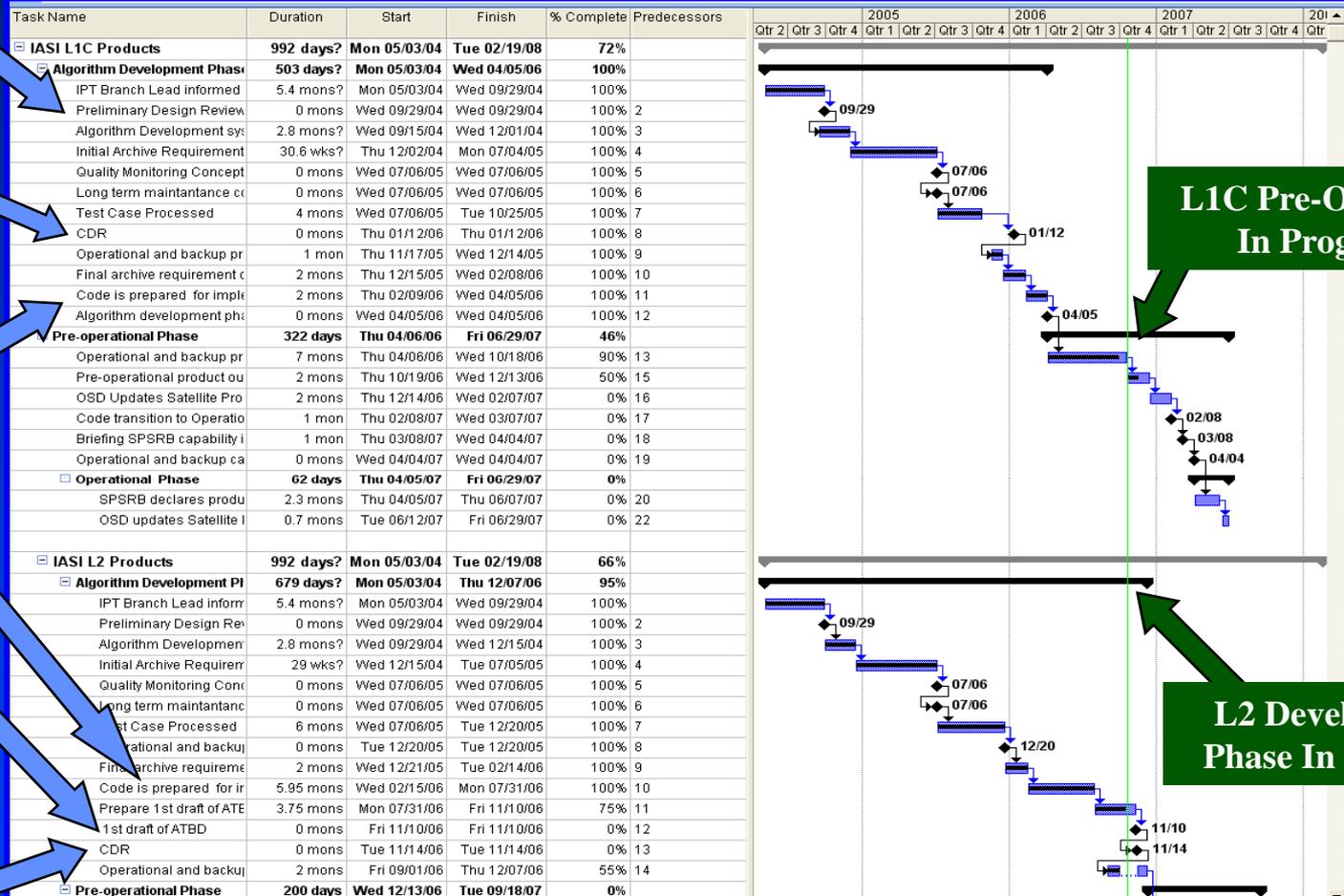


Section 1.4



Project Timeline - Design Phase

- PDR**
09/29/04
- L1C Products CDR**
01/12/06
- L1C Code**
04/05/06
- L2 Code**
07/31/06
- ATBD 1st draft**
11/10/06
- L2 Products CDR**
11/14/06



Section 1.4 - Timeline Partition



Project Plan - Changes Since Project Requirements Review

Raytheon

<Describe any changes to the project plan – objectives, stakeholders, tasks, schedule and milestones – that have occurred since the Project Requirements Review. Use multiple slides as necessary for clarity.>

<OR, if there have been no changes, state the following:>

- There have been no changes to the project plan since the Project Requirements Review

Section 1.5

Project Plan

Stakeholder Involvement (1)



- **<Describe the involvement of stakeholders in the project, noting compliance or deviation from the project plan. Use multiple slides as necessary for clarity. Follow the format shown on this slide and the next slide.>**
- **Development Lead**
 - » Sub-bullet 1 (Description of involvement related to the Project Plan)
 - »
 - » Sub-bullet M (Description of involvement related to the Project Plan)
- **Development Scientists**
 - » Sub-bullet 1 (Description of involvement related to the Project Plan)
 - »
 - » Sub-bullet M (Description of involvement related to the Project Plan)
- **Development Testers**
 - » Sub-bullet 1 (Description of involvement related to the Project Plan)
 - »
 - » Sub-bullet M (Description of involvement related to the Project Plan)

Project Plan

Stakeholder Involvement (2)



- **Development Programmers**
 - » Sub-bullet 1 (Description of involvement related to the Project Plan)
 - »
 - » Sub-bullet M (Description of involvement related to the Project Plan)
- **QA**
 - » Sub-bullet 1 (Description of involvement related to the Project Plan)
 - »
 - » Sub-bullet M (Description of involvement related to the Project Plan)
- **CM/DM**
 - » Sub-bullet 1 (Description of involvement related to the Project Plan)
 - »
 - » Sub-bullet M (Description of involvement related to the Project Plan)
- **Customers / Users**
 - » Sub-bullet 1 (Description of involvement related to the Project Plan)
 - »
 - » Sub-bullet M (Description of involvement related to the Project Plan)



CDR Guidelines and Check List

- Guidelines for the CDR reviewers are in STAR EPL process asset PRG-8.1
 - » Reviewers can access this document at <pointer to PRG-8.1>
- The CDR Review Check List is STAR EPL process asset CL-8.1
 - » Reviewers can access this document at <pointer to CL-8.1>



CDR Guidelines and Check List

- Guidelines for the CDR reviewers are in STAR EPL process asset PRG-8.1
 - » Reviewers can access this document at <pointer(s) to PRG-8.1>
- The CDR Review Check List is in the Development Project Plan (DPP) Appendix C
 - » Reviewers can access this document at <pointer(s) to DPP Appendix C>



CDR Report

- The CDR Report (CDRR) is a standard artifact of the STAR EPL process.
 - » The CDR reviewers should produce this report after conducting then CDR.
 - » The report will be a critical artifact for the Gate 4 Review.
- Guidelines for the CDRR are found in STAR EPL process asset DG-8.3
 - » CDR reviewers can access this document at <pointer to DG-8.3>



Review Objectives (1)

- **Review the project plan**
 - » Development Project Plan (DPP)
- **Review the Preliminary Design Review (PDR)**
 - » Preliminary Design Review Report (PDRR)
- **Review the operations concept**
 - » Operations Concept Document (OCD)
- **Review the requirements**
 - » Focus on changes since PDR
 - » Requirements Allocation Document (RAD)
- **Review the algorithm theoretical basis**
 - » Algorithm Theoretical Basis Document (ATBD)



Review Objectives (2)

- **Review the software system architecture and interfaces**
 - » Software Architecture Document (SWA)
- **Review the detailed design**
 - » Detailed Design Document (DDD)
- **Review the plans for quality assurance**
 - » Project Baseline Report (PBR)
 - » Verification and Validation Plan (VVP)
- **Review the requirements allocation**
 - » Focus on changes since PDR
- **Review risks and actions**



Review Objectives (3)

- <Project-Unique Objective 1>
 - » Sub-bullets
- <Project-Unique Objective 2>
 - » Sub-bullets
-
 - »
- <Project-Unique Objective N>
 - » Sub-bullets



- INTRODUCTION
- **PDR REPORT**
- OPERATIONS CONCEPT
- REQUIREMENTS
- ALGORITHM THEORETICAL BASIS
- SOFTWARE ARCHITECTURE AND INTERFACES
- DETAILED DESIGN DESCRIPTION
- QUALITY ASSURANCE
- REQUIREMENTS ALLOCATION
- RISKS AND ACTIONS
- SUMMARY AND CONCLUSIONS

Section 2 Setup Slide



Section 2 - Preliminary Design Review Report

Presented by

<Presenter's Name>

<Presenter's Title/Role>

<Presenter's Organization>



Preliminary Design Review Report (PDRR)

- PDRR is the approved report of the PDR reviewers. The PDRR can be obtained at <Pointer to the PDRR>
- The PDRR includes approval status for each preliminary design requirement
 - » Status should be Pass, Conditional Pass, Waive, or Defer
 - » Items with “Conditional Pass” status must have associated actions that should be closed prior to CDR
 - » Items with “Defer” status must have associated actions
 - Actions deferred to the CDR must be addressed prior to CDR approval
 - » Agreement of relevant stakeholders should be documented
- The PDRR includes an assessment of risk items, with recommendations for risk mitigation

Some of the risk items will be addressed in Section 10 of this CDD

Section 2.1



<Project Name> CDR - Entry Criteria

Raytheon

- <List the entry criteria for this CDR. Present as bullets. Use multiple slides as necessary for clarity. The following two slides should be used if the standard CDR entry criteria, documented in STAR EPL Checklist CL-8.1, are used.>
- <If the entry criteria for a particular project have been tailored, revise these slides as necessary to capture the set of entry criteria documented in the PDRR.>



<Project Name> CDR - Entry Criteria # 1 - 4

- **Entry # 1** - A Preliminary Design Review Report (PDRR) has been written. The CDR reviewers have access to the current baseline version of the PDRR.
- **Entry # 2** - A Development Project Plan (DPP) has been written. The CDR reviewers have access to the current baseline version of the DPP.
- **Entry # 3** - An Operations Concept Document (OCD) has been written. The CDR reviewers have access to the current baseline version of the OCD
- **Entry # 4** - A Requirements Allocation Document revision (RAD) has been written. The CDR reviewers have access to the current baseline version of the RAD.



<Project Name> CDR - Entry Criteria # 5 - 7

- **Entry # 5** - An Algorithm Theoretical Basis Document (ATBD) has been written. The CDR reviewers have access to the current baseline version of the ATBD.
- **Entry # 6** - A Software Architecture Document (SWA) has been written. The CDR reviewers have access to the current baseline version of the SWA.
- **Entry # 7** – A Detailed Design Document (DDD) has been written for each unit of the software architecture. The CDR reviewers have access to the current baseline version of each DDD.



<Project Name> CDR - Entry Criteria # 8 - 10

- **Entry # 8** - A Verification and Validation Plan (VVP) has been written. The CDR reviewers have access to the current baseline version of the VVP.
- **Entry # 9** - A Critical Design Document (CDD) has been written. CDR review objectives are clearly stated in the CDD.
- **Entry # 10** - A Project Baseline Report (PBR) has been written. The CDR reviewers have access to the current baseline version of the PBR.



<Project Name> - Tailored CDR Entry Criteria

- <List CDR entry criteria that are non-standard (added or revised from the standard set of entry criteria in STAR EPL Check List CL-8.1), explain the deviation, provide a rationale, and assess the risk, usually by reference to a risk # to be discussed in Section 10>
- <If there are no tailored entry criteria, omit this slide>



<Project Name> - Waived CDR Entry Criteria

- <List any standard entry criteria that have been waived for this CDR. Provide a rationale, based on the PDRR, and assess the risk, usually by reference to a risk # to be discussed in Section 10. Use multiple slides as necessary for clarity.>
- <If there are no waived entry criteria, omit this slide>



<Project Name> CDR - Exit Criteria

- <List the exit criteria for this CDR. Present as bullets. Use multiple slides as necessary for clarity. The following slides should be used if the standard CDR exit criteria, documented in STAR EPL Checklist CL-8.1, are used.>
- <If the exit criteria for a particular project have been tailored, revise these slides as necessary to capture the set of exit criteria documented in the PDRR.>



<Project Name> CDR - Exit Criteria # 1 - 5

- **Exit # 1** - PDR "Conditional Pass" items have been satisfactorily disposed of
- **Exit # 2** - PDR "Defer" items have been satisfactorily disposed of
- **Exit # 3** - Project plan and DPP are satisfactory
- **Exit # 4** - Operations concept and OCD are satisfactory
- **Exit # 5** - Requirements changes since PRR are approved.



<Project Name> CDR - Exit Criteria # 6 - 9

- **Exit # 6** - Algorithm theoretical basis and ATBD are satisfactory.
- **Exit # 7** - Software architecture and SWA are satisfactory
- **Exit # 8** – Software detailed design and DDDs are satisfactory
- **Exit # 9** - Verification and validation plan and VVP are satisfactory.



<Project Name> CDR - Exit Criteria # 10 - 14

- **Exit # 10** - Requirements allocation and RAD are satisfactory
- **Exit # 11** - Project baseline and PBR are satisfactory
- **Exit # 12** - The CDRR documents the current status of project risks, actions and CDR exit criteria.
- **Exit # 14** – Project risks and actions are acceptable. Project is ready for the Build phase.



<Project Name> - Tailored CDR Exit Criteria

- <List CDR exit criteria that are non-standard (added or revised from the standard set of exit criteria in STAR EPL Checklist CL-8.1), explain the deviation, provide a rationale, and assess the risk, usually by reference to a risk # to be discussed in Section 10>
- <If there are no tailored exit criteria, omit this slide>

<Project Name> - Waived CDR Exit Criteria



- <List any standard exit criteria that have been waived for this CDR. Provide a rationale and assess the risk, usually by reference to a risk # to be discussed in Section 10. Use multiple slides as necessary for clarity.>
- <If there are no waived exit criteria, omit this slide>



- INTRODUCTION
- PDR REPORT
- **OPERATIONS CONCEPT**
- REQUIREMENTS
- ALGORITHM THEORETICAL BASIS
- SOFTWARE ARCHITECTURE AND INTERFACES
- DETAILED DESIGN DESCRIPTION
- QUALITY ASSURANCE
- REQUIREMENTS ALLOCATION
- RISKS AND ACTIONS
- SUMMARY AND CONCLUSIONS

Section 3 Setup Slide



Section 3 – Operations Concept

Presented by

<Presenter's Name>
<Presenter's Title/Role>
<Presenter's Organization>



Operations Concept - Overview

- Before requirements can be developed for a product and product system, the developers must know the intentions of the customers and/or users of the product. They must have the answers to the following questions:
 - » Why is this product being produced?
 - » How will this product be used?
 - » How should this product be produced (operational scenario)?
- The answers to the preceding questions should be derived from customer/user needs and expectations
 - » Often, the customers/users will have documented their concept of operations in a ConOps document (see next slide)
 - » Prior to the Development Phase, algorithm developers should interact with the potential customers/users to produce an initial algorithm theoretical basis that is consistent with their concept of operations
- Given the required input from customers/users, the development team should develop and document timeline scenarios for product operation and user interaction, at a level of detail and maturity appropriate for each step in the Development Phase
- The operations concept is typically refined by the development team, in consultation with customers/users, as the product solutions and design are matured through the Design Development phase.



Operations Concept Document

- Operations Concept Document (OCD)
 - » Describes how the users' vision can be realized in an operational environment. Guidelines in STAR EPL process asset DG-6.1 <Pointer to DG-6.1>
- OCD v1r0 (PRR artifact), the initial version, should capture customer needs and expectations
- OCD v1r1 (PDR artifact) should provide operational scenarios for product operation and user interaction for each alternative solution under consideration at PDR.
- **OCD v1r2, a CDR artifact, is available at <pointer to OCD v1r2>**
 - » **Documents the refinement of the operations concept that may occur as a result of detailed design development**



Why Are The Products Being Produced?

- <Itemize customer/user needs>
 - » <Refer to a customer ConOps, if one exists>
 - » <Explain how this question was answered, if a ConOPs does not exist>
 - » <This should be documented in the OCD, consistently with the DPP. Use text, figures, tables from the OCD>
 - » <Use multiple slides as necessary for clarity>
 - » <Most of this material should be obtainable from the Project Requirements Document (PRD) Section 3 slides, but there may be additional material obtainable
- Section 3.2** CD v1r2 update.>



How Will The Products Be Used?

- <Itemize customer/user expectations>
 - » <Refer to a customer ConOps, if one exists>
 - » <Explain how this question was answered, if a ConOPs does not exist>
 - » <This should be documented in the OCD, consistently with the DPP. Use text, figures, tables from the OCD>
 - » <Use multiple slides as necessary for clarity>
 - » <Most of this material should be obtainable from the Project Requirements Document (PRD) Section 3 slides, but there may be additional material obtainable from the OCD v1r2 update.>



How Should The Products Be Produced? (1)

- Available production environments
 - » <Describe the production environments that are available for the product lifecycle, including development, transition, operations and delivery>
 - <Itemize the hardware, software and personnel resources that can be available for each environment>
 - <Describe how the environments can be integrated. Include boundaries and constraints>
 - » <Describe the production environments that are available for the product lifecycle, including development, transition, operations and delivery>
 - » <This should be documented in the OCD. Use text, figures, tables from the OCD. Use multiple slides as necessary for clarity>
 - » <Some of this material should be obtainable from the Project Requirements Document (PRD) Section 3 slides, but there will typically be additional material obtainable from the OCD v1r2>



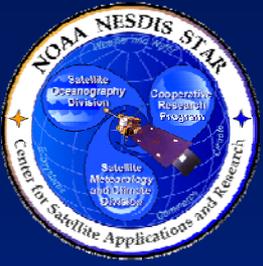
How Should The Products Be Produced? (2)

- Production and Delivery scenarios
 - » < Describe production and delivery scenarios, consistent with the level of detail in the customer's concept of operations, the production environment constraints, and operator needs and expectations.>
 - <A scenario is a sequence of events that might occur in the production and use of the product, which is used to make explicit the needs of the stakeholders>
 - » <This should be documented in the OCD. Use text, figures, tables from the OCD. Use multiple slides as necessary for clarity>
 - » <Some of this material should be obtainable from the Project Requirements Document (PRD) Section 3 slides, but there will typically be additional material obtainable from the OCD v1r2 update.>



- INTRODUCTION
- PDR REPORT
- OPERATIONS CONCEPT
- **REQUIREMENTS**
- ALGORITHM THEORETICAL BASIS
- SOFTWARE ARCHITECTURE AND INTERFACES
- DETAILED DESIGN DESCRIPTION
- QUALITY ASSURANCE
- REQUIREMENTS ALLOCATION
- RISKS AND ACTIONS
- SUMMARY AND CONCLUSIONS

Section 4 Setup Slide



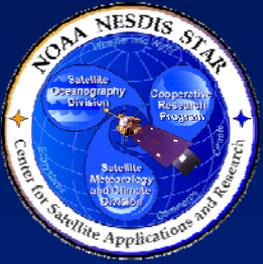
Section 4 - Requirements

Presented by

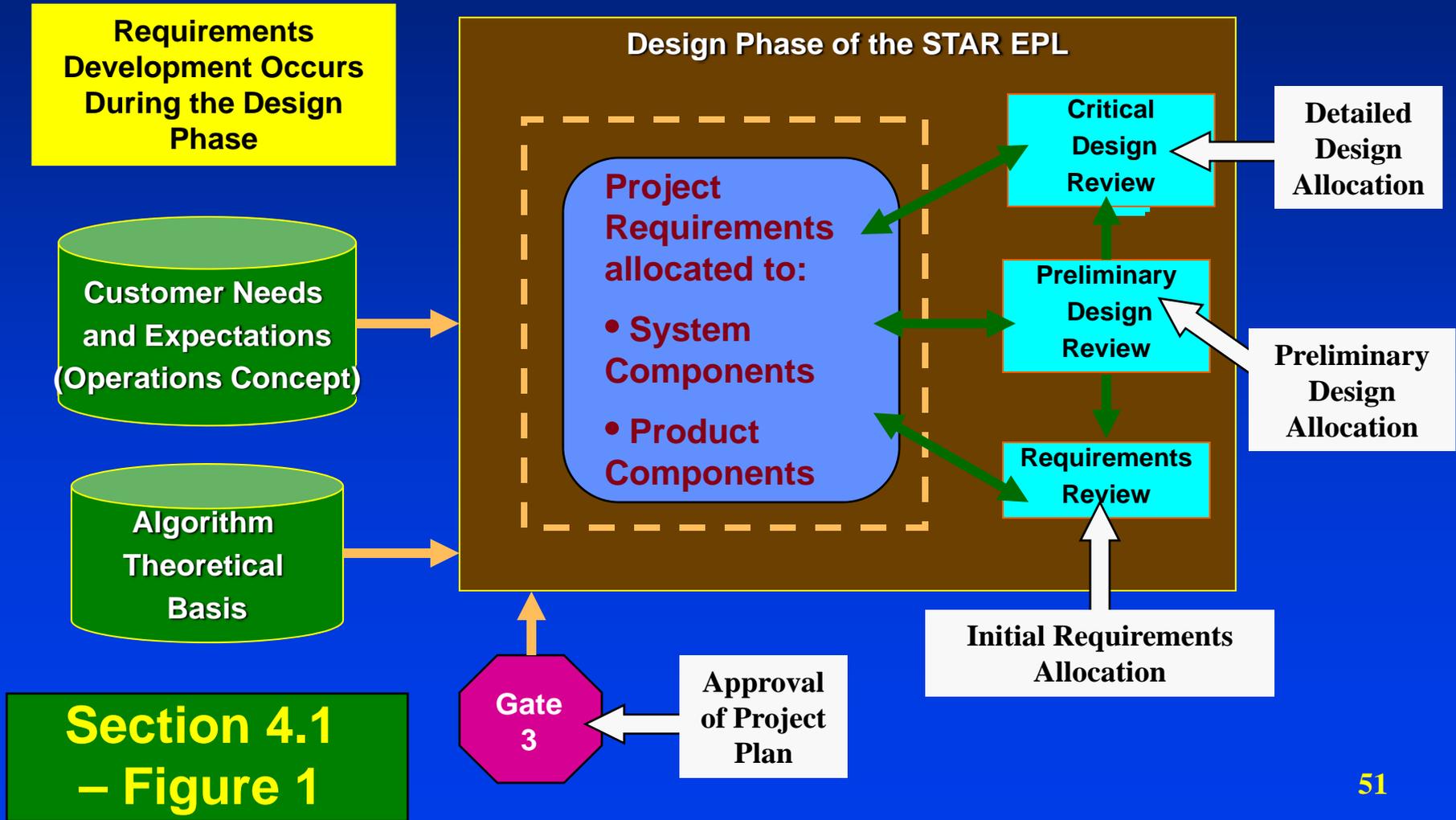
<Presenter's Name>

<Presenter's Title/Role>

<Presenter's Organization>

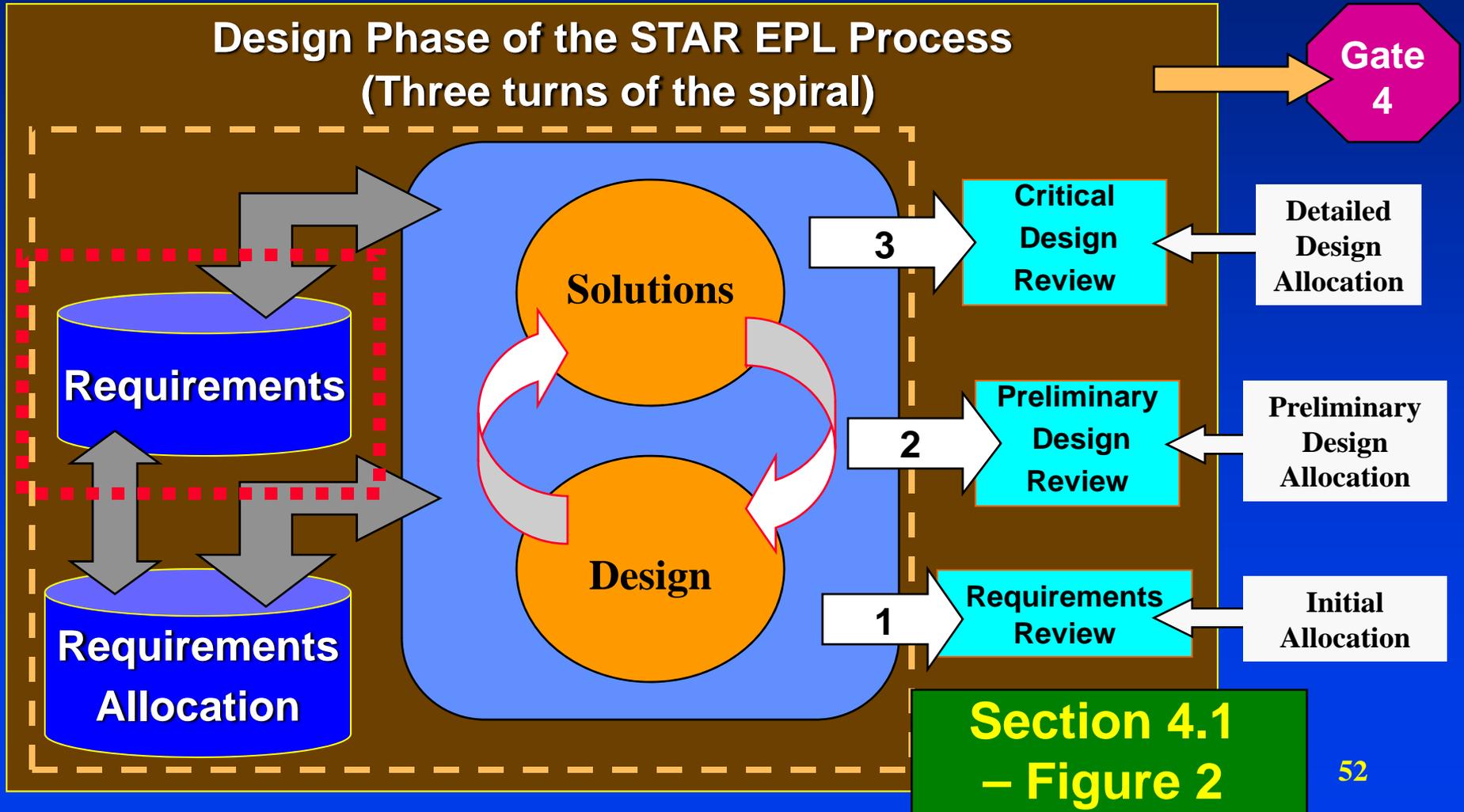


Requirements Development Process





Iterative (Spiral) Development of Requirements





Project Requirements Have Been Established and Refined

- **Established at Project Requirements Review (PRR)**
 - » Project Requirements Document (PRD)
 - » Project Requirements Review Report (PRRR)
 - » Requirements Allocation Document (RAD) v1r0
- **Refined for Preliminary Design Review (PDR)**
 - » Preliminary Design Document (PDD)
 - » Preliminary Design Review Report (PDRR)
 - » Requirements Allocation Document (RAD) v1r1
- **Refined for Critical Design Review (CDR)**
 - » Critical Design Document (CDD – this presentation)
 - » **Requirements Allocation Document (RAD) v1r2**



Requirements Allocation Document

Raytheon

- Requirements Allocation Document (RAD)
 - » RAD v1r2, a CDR artifact, can be obtained at <Pointer to RAD v1r2>
 - » RAD Document Guidelines in STAR EPL process asset DG-6.2 <Pointer to DG-6.2>
- The RAD contains the basic and derived requirements for the work products.
 - » **RAD v1r2 notes updates to requirements after PDR**
- RAD v1r2 includes the allocation of the requirements to system components and product components, at a detailed design level of detail
 - » **Requirements allocation will be addressed in Section 8 of this CDD**

Section 4.2



New Requirements Since PDR

- **<List each new requirement>**
 - » <If a derived requirement, list higher-level driving requirements>
 - » <If a basic requirement, list new derived requirements>
 - » <Note whether the new requirement has been approved at a delta Requirements Review>
 - » <If the new requirement has not been approved:>
 - <Explain rationale for the new requirement (e.g., revealed by detailed design issue, new customer request, etc.)>
 - <Note potential effects on the project plan>
 - <Document the agreement of affected stakeholders>
 - <Note new or modified risks that result from the new requirement>
 - <Note any recommended actions that result from the new requirement>



Requirements Changes Since PDR

- **<List each requirement change>**
 - » <If a derived requirement, list higher-level driving requirements>
 - » <If a basic requirement, list derived requirements that are affected>
 - » <Note whether the change has been approved at a delta Requirements Review>
 - » <If the change has not been approved:>
 - <Explain rationale for the change (e.g., revealed by detailed design issue, operational constraint)>
 - <Note potential effects on the project plan>
 - <Document the agreement of affected stakeholders>
 - <Note new or modified risks that result from the change>
 - <Note any recommended actions that result from the change>



- INTRODUCTION
- PDR REPORT
- OPERATIONS CONCEPT
- REQUIREMENTS
- **ALGORITHM THEORETICAL BASIS**
- SOFTWARE ARCHITECTURE AND INTERFACES
- DETAILED DESIGN DESCRIPTION
- QUALITY ASSURANCE
- REQUIREMENTS ALLOCATION
- RISKS AND ACTIONS
- SUMMARY AND CONCLUSIONS

Section 5 Setup Slide



Section 5 – Algorithm Theoretical Basis

Presented by

<Presenter's Name>
<Presenter's Title/Role>
<Presenter's Organization>



Algorithm Theoretical Basis

- Purpose: Provide product developers, reviewers and users with a theoretical description (scientific and mathematical) of the algorithm that is used to create a product that meets user requirements.
- Documented in the Algorithm Theoretical Basis Document (ATBD)
 - » ATBD Guidelines in DG-1.1 <pointer to DG-1.1>
 - » The ATBD is developed throughout the STAR EPL, beginning with Basic Research (v1r0) and continuing through the Build phase (v2r2)
 - » The version of the ATBD developed for CDR is v2r1
 - » ATBD v2r1, a CDR artifact, is available at <Pointer to ATBD v2r1>



Algorithm Objectives

- <Describe the objectives of the algorithm, including the intended output data products and their intended use>
- <Show how the algorithm objectives are derived from the operations concept>



Sensor Inputs

- <Describe the attributes of the sensing system(s) used to supply data for the retrieval algorithm. Provide references to the appropriate Sensor Specifications and sensor review documents if available.
- <Use relevant slides from sensor reviews if available. Use multiple slides as necessary for clarity.>



Ancillary Inputs

- < Describe the attributes of all input data used by the algorithm, including ancillary data, forward models and look-up tables. Use multiple slides as necessary for clarity.>



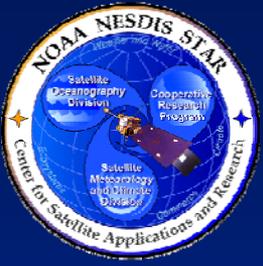
Retrieval Strategy

- <Describe the fundamental approach for retrieval at a level of detail sufficient for reviewers to determine that the algorithm theoretical description is adequate.>



Processing Outline

- <Describe the processing outline of the retrieval algorithm. All key elements and sub-elements needed to convey a comprehensive sense of the algorithm should be included. A process flow chart is recommended.>



Physical Description

- <Describe the physics and associated phenomenology key to the retrieval>
 - » <Refer to the relevant section of the ATBD>
 - » <Note whether any part of the physical description is not based on proven algorithm heritage – in that case, evaluate the risk of new algorithm physics.>



Mathematical Description

- <Describe the mathematics used by the retrieval, including all simplifications, approximations, and numerical methods. Refer to the relevant section of the ATBD.>



Algorithm Output

- <Describe the algorithm output, mapping output characteristics to product requirements. Refer to the relevant section of the ATBD.>



Performance Estimates

- <Describe the predicted algorithm performance and quality of the products derived from analysis and tests with simulated and/or proxy test data. Use multiple slides as necessary for clarity.>
 - » <Refer to the relevant sections of the ATBD>
 - » <Note verification methods and assumptions>
 - <Identify error sources to be tested (e.g. calibration, sensor noise, etc.)>
 - <Identify and justify stratifications to be tested, including typical and stressing conditions>
 - <Identify test data sets that span the stratifications>
 - <Describe all assumptions that have been made concerning the performance estimates. To the extent possible, the potential for degraded performance should be explored, along with mitigating strategies.>
 - » <Use text, figures etc. from the ATBD>
 - » <Include error budgets if available>
 - » <Compare performance estimates with product requirements. Identify performance risks.>



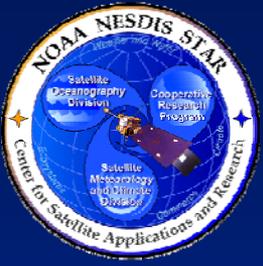
Practical Considerations

- <Describe how the algorithm is numerically implemented, including any possible issues with computationally intensive operations (e.g., large matrix inversions).>
- <Describe any important programming and procedural aspects related to implementing the numerical model into operating code.>
- <Refer to the relevant ATBD section>



- INTRODUCTION
- PDR REPORT
- OPERATIONS CONCEPT
- REQUIREMENTS
- ALGORITHM THEORETICAL BASIS
- **SOFTWARE ARCHITECTURE AND INTERFACES**
- DETAILED DESIGN DESCRIPTION
- QUALITY ASSURANCE
- REQUIREMENTS ALLOCATION
- RISKS AND ACTIONS
- SUMMARY AND CONCLUSIONS

Section 6 Setup Slide



Section 6 – Software Architecture and Interfaces

Presented by

<Presenter's Name>
<Presenter's Title/Role>
<Presenter's Organization>



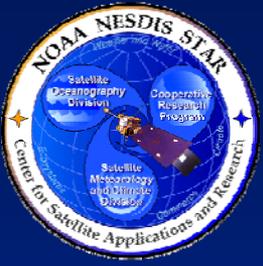
Software Architecture

- The software system is an integrated collection of software elements, or code, that implements the algorithm, producing well-defined output products from a well-defined set of input data.
- The software architecture describes the structure of the system software elements and the external and internal data flows between software elements.
- This section describes the software architecture for the preferred solution (algorithm) identified in Section 5 of this CDD.

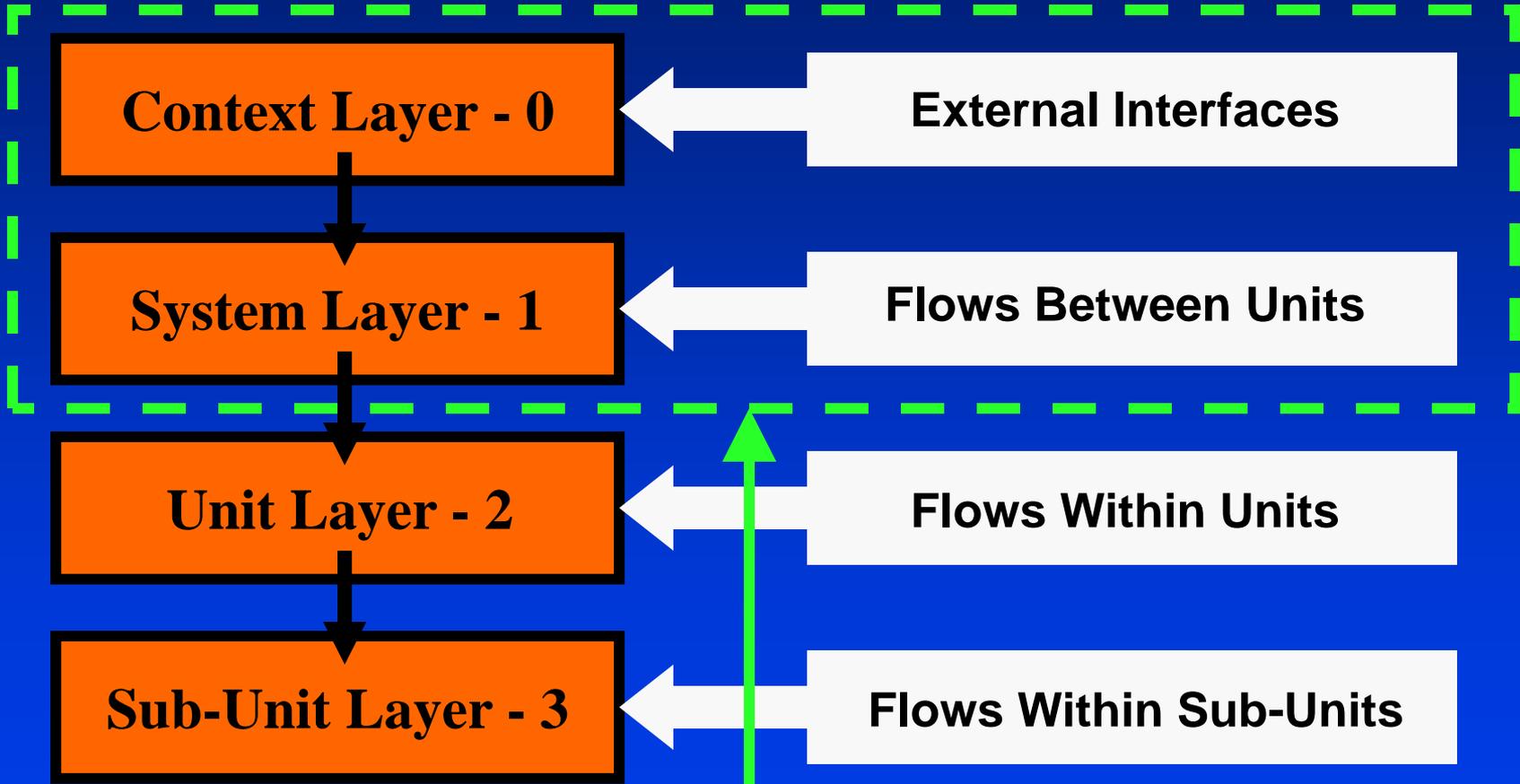


Software Architecture Documentation

- The software architecture is described in the Software Architecture Document (SWA)
 - » SWA Guidelines in STAR EPL process asset DG-1.2 <pointer to DG-1.2>
 - » The SWA is developed throughout the STAR EPL, beginning with Basic Research (v1r0) and continuing through the Build phase (v2r3)
 - » SWA v2r1, a CDR artifact, is available at <Pointer to SWA v2r1>
 - » Purpose: Demonstrate that the algorithm process flow provides for an implementation that is consistent with the mathematical basis and meets requirements.

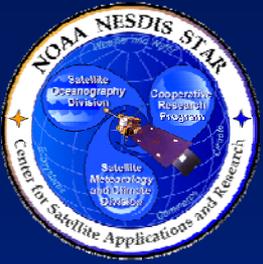


Software Architecture at PDR

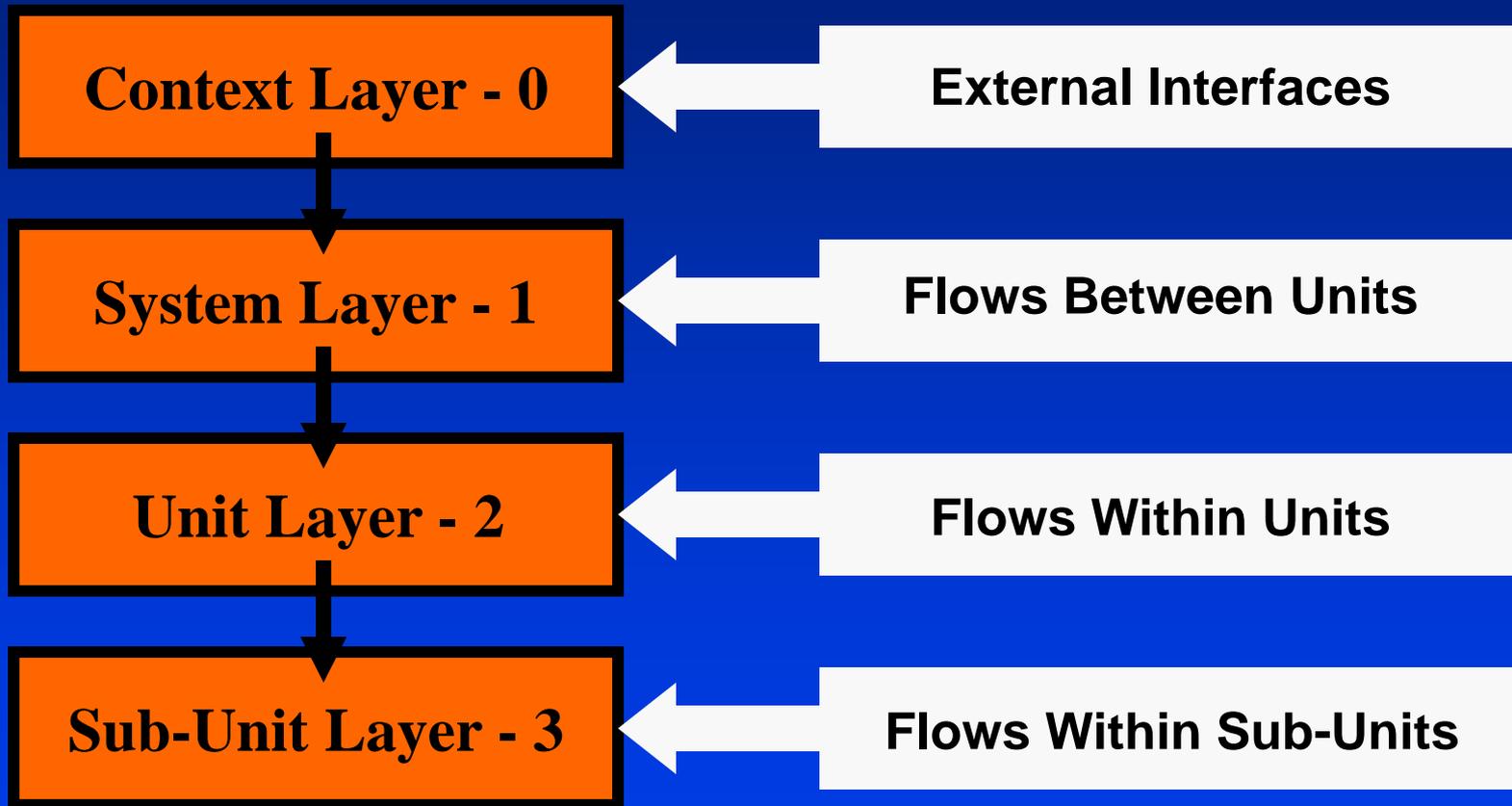


**Section 6.1
– Figure 1**

Preliminary Design Layers



Software Architecture at CDR



**Section 6.1
– Figure 2**



The Context-Layer

- The Context-Layer describes the flows between the system and its external interfaces
- An external input is defined as a data source needed by the system that is produced or made available by a process external to the system
- An external output is defined as a data sink that is produced by the system for an external user
- External interfaces must meet standard criteria to be included in the system architecture



External Interfaces - Criteria

- <State the criteria for the design of interfaces (e.g. security, acceptance testing, etc. Use multiple slides as necessary for clarity.>
 - » <Note any deviations from the STAR standard criteria>
 - <Explain any deviations>

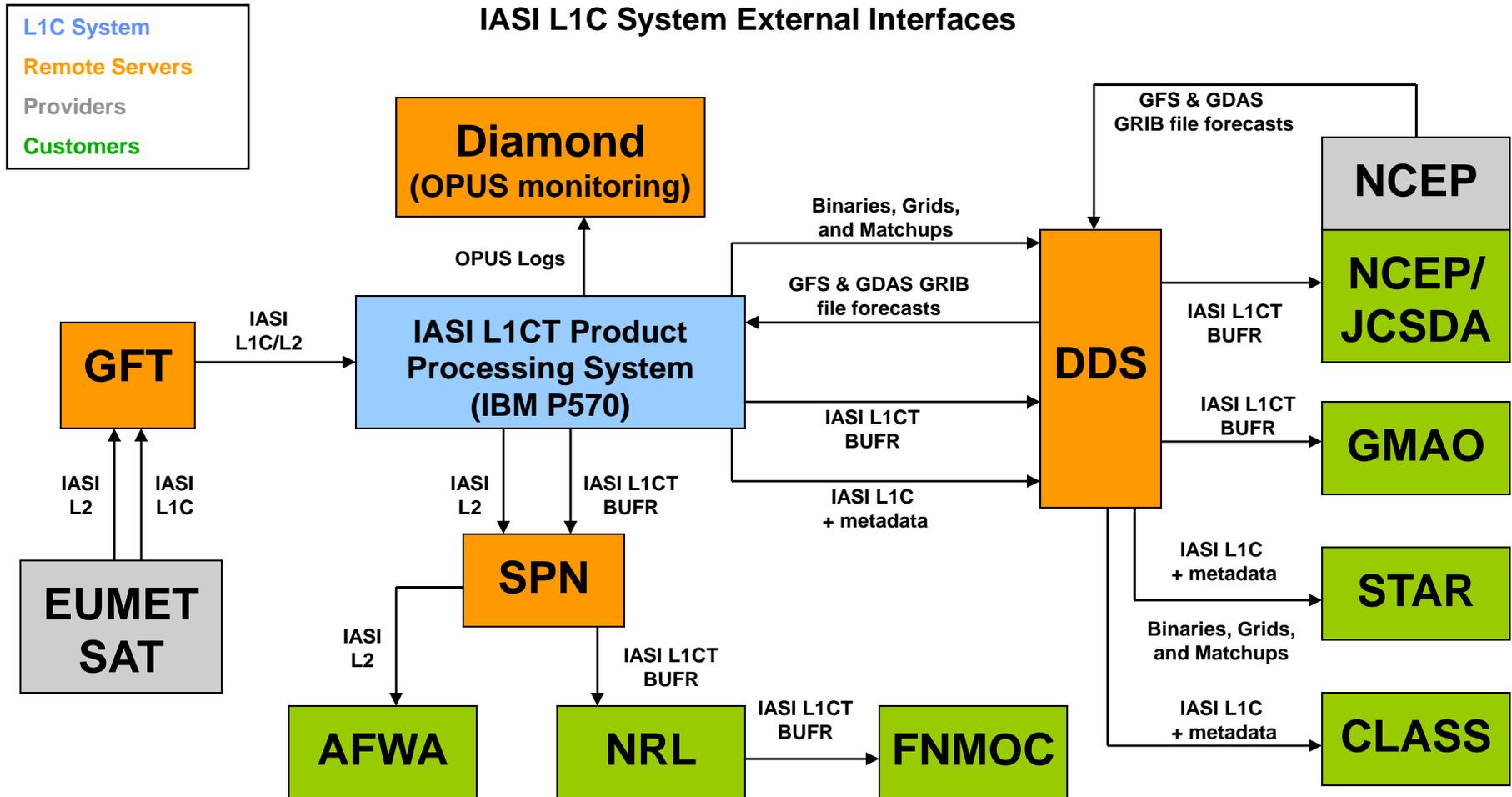


External Interface Design

- <Show all external inputs and outputs to and from the software system>
 - » Start with a context diagram (see next slide for an example). The context diagram slide may be obtained from PDD Section 6, unless there have been changes since PDR; in that case, obtain the updated diagram from the revised SWA.>



System External Interfaces



Section 6.2 – Figure 1



External Interfaces

- <Discuss each interface item at a level of detail that is warranted for the item. Adopt PDD Section 6 slides, if the external interfaces have not been changed since PDR. If there have been changes, revise the PDR slides as needed.>
- < Include a table if that adds clarity (see the next slide for an example). Alternatively, use a bulleted list.>



External Interfaces

Input and Output Context-Layer Data Flows

Interface Item	Interface Type	Source	Description
Total Column Ozone	Input	NCEP	NCEP total column ozone, re-gridded to the moderate resolution VIIRS swath
Calibrated Radiance	Input	VIIRS	VIIRS calibrated radiance in bands I1, I2, and I3, at the imagery resolution VIIRS swath
VIIRS Snow Binary Map	Output	VIIRS Snow/Ice System	Snow binary map, at the imagery resolution VIIRS swath, with associated metadata, Earth location, and quality flags



External Interfaces

- **Total Column Ozone (Input)**
 - » Source – NCEP
 - » Description - NCEP total column ozone, re-gridded to the moderate resolution VIIRS swath
- **Calibrated Radiance (Input)**
 - » Source – VIIRS
 - » Description - VIIRS calibrated radiance in bands I1, I2, and I3, at the imagery resolution VIIRS swath
- **VIIRS Snow Map (Output)**
 - » Sink – DDS
 - » Description - Snow binary map, at the imagery resolution VIIRS swath, with associated metadata, Earth location, and quality flags



External Interfaces Satisfy Acceptance Criteria

- <Demonstrate that each external interface satisfies acceptance criteria>
- <This demonstration can include analysis, simulation, security certification, and past practices>
- <Adopt slides from PDD Section 6, if the external interfaces have not changed since PDR; if there have been changes, revise the PRD slides as needed.>
- <Use figures and tables as warranted. Use multiple slides as needed for clarity.>



The System-Layer

- The System-Layer data flow expands upon the Context-Layer data flow, showing the first layer of decomposition.
 - » In addition to the System-Layer inputs and outputs, the major processing units are shown along with their inputs and outputs.
 - » Each unit is designed as a stand-alone program for ease of testing and integration into a System-Layer scheduler.
- <Show the System-Layer data flows as a data flow diagram (see next slide for an example). Adopt the PRD Section 6 diagram, if the data flows are unchanged since PDR, or revise the PRD diagram as necessary.>



System-Layer Data Flows

- <Discuss each System-Layer data item at a level of detail that is warranted for the item. Adopt PDD Section 6 slides, if the external interfaces have not been changed since PDR. If there have been changes, revise the slides as needed.>

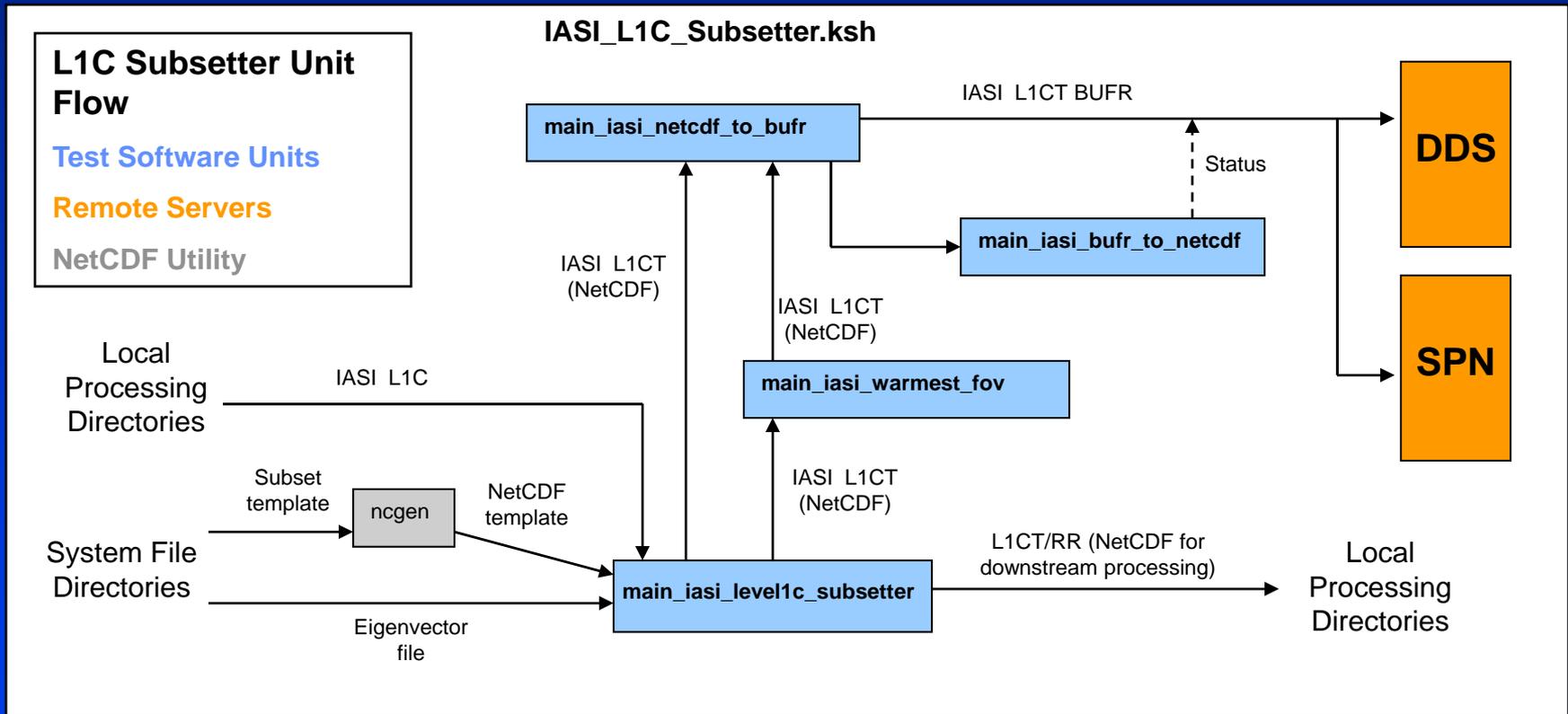


The Unit-Layer

- The Unit-Layer Data Flow decomposes the system level software architecture to the next (unit) level.
 - » In this layer, the data flows within units are described.
- <Show the unit level data as a data flow diagram (see next slide for an example). Adopt the PDD Section 6 diagram, if the data flows are unchanged since PDR, or revise the PDD diagram as necessary. The diagram should be found in SWA v2r1.>



L1C Subsetter Unit Flow



Section 6.4 – Figure 1



<Unit Name> Unit-Layer Data Flows

Raytheon

- <For each unit in turn, discuss its Unit-Layer data item at a level of detail that is warranted for the item. Use multiple slides as necessary for clarity.>



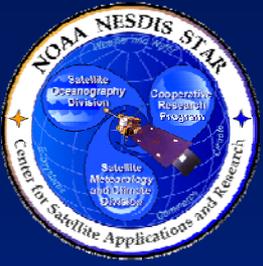
Sub-Unit-Layer Data Flows

- The Sub-Unit Layer Data Flow decomposes the unit level software architecture to the next (sub-unit) layer.
 - » At this layer, the data flows within sub-units are described.
- <Show the Sub-Unit-Layer data flows for each unit as a series of data flow diagrams. These should be adopted from SWA v2r1.>
- <Sub-Unit-Layer descriptions are usually too numerous to be effectively presented in the CDD slides. Instead, refer the reviewers to the complete description in the SWA.>



- INTRODUCTION
- PDR REPORT
- OPERATIONS CONCEPT
- REQUIREMENTS
- ALGORITHM THEORETICAL BASIS
- SOFTWARE ARCHITECTURE AND INTERFACES
- **DETAILED DESIGN DESCRIPTION**
- QUALITY ASSURANCE
- REQUIREMENTS ALLOCATION
- RISKS AND ACTIONS
- SUMMARY AND CONCLUSIONS

Section 7 Setup Slide



Section 7 - Detailed Design Description

Presented by

<Presenter's Name>

<Presenter's Title/Role>

<Presenter's Organization>



Detailed Design

- The detailed design builds on the software architecture by providing a detailed description of each system element that is defined in the software architecture
 - » Its purpose is to describe the product design at a level of detail that is sufficient for development programmers to write fully functional pre-operational code



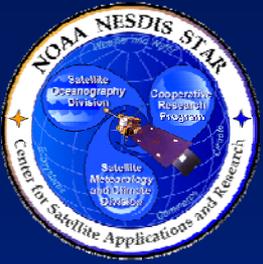
Software Detailed Design

- The software detailed design describes the project system's software functionality and design characteristics at a level of detail that covers, for each software element:
 - » Its purpose
 - » Allocation to project requirements
 - » External interfaces
 - » Decomposition into sub-elements
 - » Functional sequence
 - » Design Language
 - » Input and Output File Descriptions



Software Detailed Design Documentation

- The detailed design of each software unit is described in a Detailed Design Document (DDD)
 - » A separate DDD for each software unit identified in the System-Layer software architecture
 - » DDD Guidelines in STAR EPL process asset DG-8.1 <pointer to DG-8.1>
 - » DDD v1r0, a CDR artifact, is available at <Pointer to DDD v1r0>
- The DDD's detailed focus on code component design and design language is intended for Development Programmers, but can also be useful as reference material for operations and maintenance programmers



<Unit 1> Unit Description (1)

- <Bullets, and possibly sub-bullets, that explain the following:>
 - » <Purpose of the unit>
 - » <Requirements Allocation (i.e., list all requirements (basic and derived) that the unit satisfies; note whether the requirement is satisfied completely or partially by the unit.>
 - » <Interfaces – list all interfaces, including external interfaces and system internal interfaces. Refer to SWA and DDD for material.>
 - » <Functional Sequence – List the major processing steps performed by the unit>
 - » <Design Language – Note that the DDD includes the unit design language in DDD section <X>.
 - » <List assumptions and limitations that apply to the unit design>
- <Use multiple slides, figures, tables as warranted. Obtain content from the DDD.>



<Unit 1 Name> Unit - Sub-Unit Descriptions

- The <Unit 1 Name> unit includes the following sub-units:
 - » <Sub-Unit 1 Name>
 - » <.....>
 - » <Sub-Unit N Name>
- For each of these sub-units, the detailed design description includes:
 - » Purpose
 - » Requirements allocation
 - » Interfaces
 - » Functional sequence
 - » Design language
 - » Assumptions and limitations



<Unit 1 Name> Unit - <Sub-Unit Name> Sub-Unit Description

- <Provide a description for each sub-unit of unit 1. Use bullets, and possibly sub-bullets, that explain the following:>
 - » <Purpose of the sub-unit>
 - » <Requirements Allocation (i.e., list all requirements (basic and derived) that the sub-unit satisfies; note whether the requirement is satisfied completely or partially by the unit.>
 - » <Interfaces – list all interfaces, including external interfaces (controlled by an ICD) and system interfaces. Refer to SWA and DDD for material.>
 - » <Functional Sequence – List the major processing steps performed by the sub-unit>
 - » <Design Language – Note that the DDD includes the sub-unit design language in DDD section <X>.
 - » <List assumptions and limitations that apply to the sub-unit design>
- <Use multiple slides, figures, tables as warranted. Obtain content from the DDD.>
- <Repeat for each sub-unit of Unit 1>



<Unit 2> Unit Description (1)

- <Bullets, and possibly sub-bullets, that explain the following:>
 - » <Purpose of the unit>
 - » <Requirements Allocation (i.e., list all requirements (basic and derived) that the unit satisfies; note whether the requirement is satisfied completely or partially by the unit.)>
 - » <Interfaces – list all interfaces, including external interfaces and system internal interfaces. Refer to SWA and DDD for material.>
 - » <Functional Sequence – List the major processing steps performed by the unit>
 - » <Design Language – Note that the DDD includes the unit design language in DDD section <X>.
 - » <List assumptions and limitations that apply to the unit design>
- <Use multiple slides, figures, tables as warranted. Obtain content from the DDD.>



<Unit 2 Name> Unit - Sub-Unit Descriptions

- The <Unit 2 Name> unit includes the following sub-units:
 - » <Sub-Unit 1 Name>
 - » <.....>
 - » <Sub-Unit N Name>
- For each of these sub-units, the detailed design description includes:
 - » Purpose
 - » Requirements allocation
 - » Interfaces
 - » Functional sequence
 - » Design language
 - » Assumptions and limitations



<Unit 2 Name> Unit - <Sub-Unit Name> Sub-Unit Description

- <Provide a description for each sub-unit of unit 1. Use bullets, and possibly sub-bullets, that explain the following:>
 - » <Purpose of the sub-unit>
 - » <Requirements Allocation (i.e., list all requirements (basic and derived) that the sub-unit satisfies; note whether the requirement is satisfied completely or partially by the unit.>
 - » <Interfaces – list all interfaces, including external interfaces (controlled by an ICD) and system interfaces. Refer to SWA and DDD for material.>
 - » <Functional Sequence – List the major processing steps performed by the sub-unit>
 - » <Design Language – Note that the DDD includes the sub-unit design language in DDD section <X>.
 - » <List assumptions and limitations that apply to the sub-unit design>
- <Use multiple slides, figures, tables as warranted. Obtain content from the DDD.>
- <Repeat for each sub-unit of Unit 2>



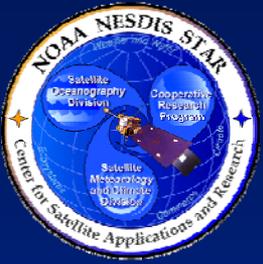
<Unit N Name> Unit - <Sub-Unit Name> Sub-Unit Description

- <Repeat the unit and sub-unit descriptions for each of the other units>



Look Up Table Descriptions

- <Explain the purpose and function of each LUT in the algorithm design>
- <List and describe each entry in the LUT. Description should include the data format. Refer to the relevant subsection of DDD Section 4> OR <Include a table for each LUT, taken from the DDD. An example is shown on the next slide.>
- <Repeat for each LUT>



<Look Up Table Name> Look Up Table

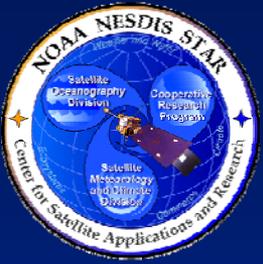
Parameter	Dimension	Values	Units
band	Float	1, 2, 3, 4, 5, 6, 7, 8, or 9	None
snow_type	Float	1, 2, 3, ..., or 24.	None
snow_grain_size	Float	50, 100, 250, 500, 750, or 1000	microns
snow_impurity	Float	0, 1, 10, or 100	ppmw
aot_bins	Float (4)	0., 0.2366, 0.5472, 1.0	None
wvot	Float(3)	0.0, 0.4323, 2.0	gm/cm ²
oot	Float(3)	0.0, 0.1967, 0.5	cm-atm
sza_bins	Float(11)	Equally spaced in cos(sza) from sza = 85:0 degrees	None
liza_bins	Float (11)	Equally spaced in cos(liza) from liza = 85:0 degrees	None
sva_bins	Float(7)	Equally spaced in cos(sva) from sva = 70:0 degrees	None
lsva_bins	Float (10)	Equally spaced in cos(lsva) from lsva = 85:0 degrees	None
relaz_bins	Float (7)	0, 30, 60, 90, 120, 150, 180	Degrees
snow_toa_reflectance	Float (n_aot, n_wvot, n_oot, n_sza, n_liza, n_sva, n_lsva, n_relaz)	From RTM	None

**Section
7.4 –
Table 1**



File Descriptions

- <Explain the purpose and function of each file in the algorithm design, except for source-code files and LUT files (these were described in Sections 7.3 and 7.4). Files can be parameter files, system control files, input/intermediate/output data files, and ancillary data files.>
- <List and describe the contents of each file. Description should include the data format. Refer to the relevant subsection of DDD Section 4> OR <Include tables for each file, taken from the DDD. An example is shown on the next slide.>
- <Repeat for each file>



<File Name> - File Contents and Format (1)

SNOW COVER EDR SNOW BINARY MAP		
FIELD Name	Data Type	HDF Dimension Names
"BinaryMap"	Byte	2-dimensional array of integers with dimensions (Number of imagery resolution pixels in cross-track dir, Number of imagery resolution pixels in along-track dir)
<p>Description: 8-bit byte array containing "1" for each pixel classified as "snow" and "0" for each pixel classified as "no snow".</p> <p>FIELD Attributes: Long Name: "VIIRS Snow Cover EDR Snow Binary Map" Units: none Valid Range: 0 or 1 Fill Value: 255</p>		

**Section 7.5 –
Table 1**



- INTRODUCTION
- PDR REPORT
- OPERATIONS CONCEPT
- REQUIREMENTS
- ALGORITHM THEORETICAL BASIS
- SOFTWARE ARCHITECTURE AND INTERFACES
- DETAILED DESIGN DESCRIPTION
- **QUALITY ASSURANCE**
- REQUIREMENTS ALLOCATION
- RISKS AND ACTIONS
- SUMMARY AND CONCLUSIONS

Section 8 Setup Slide



Section 8 - Quality Assurance

Presented by

<Presenter's Name>
<Presenter's Title/Role>
<Presenter's Organization>



Quality Assurance

- Quality assurance consists of PROCESS QA and PRODUCT QA
- PROCESS QA is concerned with assuring that the STAR EPL process standards are met during the planning, development, operations, and distribution phases of the project.
 - » Process QA is achieved through the standard reviews of the STAR EPL process. Each review checklist and entry/exit criteria are designed to ensure that the relevant process standards are met by the implementation of standard practices during the steps leading up to the review.
- PRODUCT QA is concerned with assuring that the work products created during the project's lifecycle meet their requirements
 - » Product QA is achieved by verification of the project's work products and validation of the products, operator needs, and user needs



Configuration Management Stakeholders

- <Identify the CM/DM stakeholders for the project. Verify their commitment to the Project Plan.>
 - » <This material should have been presented at the PDR. Adopt the PDR slides and update them if necessary to capture changes since PDR.>



Configuration Management Tools

- <Describe the CM/DM tools that are in use for the project. Use multiple slides as necessary for clarity.>
 - » <This material should have been presented at the PDR. Adopt the PDR slides and update them if necessary to capture changes since PDR.>



Project Baseline Report

- The project's baseline and change history is maintained in a Project Baseline Report (PBR).
 - » Document guidelines are in STAR EPL process asset DG-5.4.
 - <Pointer to DG-5.4>
 - » The PBR includes the change history, approval status, and location of every Configuration Item in the project's baseline.
 - » PBR v2r4, a CDR artifact, can be accessed at <pointer to PBR v2r4>



Verification and Validation Plan

- Verification is the formal process of confirming that the requirements specified for a specific product or system are satisfied by the completed product or system.
- Validation is a process of evaluation, integration, and test activities conducted to ensure that the final developed system will satisfy the needs and expectations of customers, users, and operators
- In a well-designed system, needs and expectations are completely captured by the requirements allocation (see Section 9 of this CDD) – in that case, there is no meaningful distinction between verification and validation
- The methods and planned activities for verification and validation of the project's process and products constitutes the project verification and validation plan



Verification and Validation Documentation

- The plan for verification and validation is documented in a Verification and Validation Plan (VVP)
- VVP v1r0 (PRR artifact), the initial version, provided an initial plan for verification and validation of the requirements identified in RAD v1r0
- VVP v1r1 (PDR artifact) updated the plan to reflect new or changed requirements and requirements allocations resulting from the preliminary design development (RAD v1r1)
- **VVP v1r2, a CDR artifact, is available at <pointer to VVP v1r2>**
 - » Updates the plan to reflect new or changed requirements and requirements allocations resulting from the detailed design development (RAD v1r2)



Verification Items

- <Identify the work products to be verified. These should be traceable to the software architecture. Note that this will be documented in the project's VVP.>
- <Identify the requirements to be satisfied by each work product selected for verification. Note that these are documented in the RAD, to be discussed in the next section.>
- <This material should have been presented at the PDR. Adopt the PDR slides and update them to capture updates in VVP v1r2.>



Verification Methods

- <Describe the verification methods that will be used, as described in the VVP>
 - » The verification methods should be described in as much detail as possible and a rationale for their selection should be provided.>
 - » Use figures, graphs, examples as warranted. Use multiple slides as necessary for clarity and completeness.>
- <Note which verification items will be verified with each method or combination of methods. If available, show a matrix obtained from the VVP.>
- <This material should have been presented at the PDR. Adopt the PDR slides and update them to capture updates in VVP v1r2.>



Verification in the Project Plan

- <Show the project plan with verification tasks highlighted>
- <Extract (as bullets) the verification task name, start date, end date, predecessor tasks, and successor tasks from the project plan>
- <Note any adjustments in the DPP (usually, schedule and resources) that are needed to accommodate the updated verification plan. For example, new verification tasks or a revised time interval for one or more verification tasks, with consequent impact on successor tasks. Most important: note any risk to milestone dates and impact on successor milestones.>
- These risks will be addressed in a later section (“Risks”) of this presentation
- <This material should have been presented at the PDR. Adopt the PDR slides and update them to capture changes to the project plan.>



Validation of Products

- <Identify user-driven requirements on the product or products to be validated. These are typically found in the project's basic requirements, should be documented in the RAD, and should have been discussed in Section 4 ("Requirements") of this presentation.>
- <For each product component, describe the scope of the validation (e.g., product quality attributes, validation environments, validation campaigns). Distinguish between pre-launch and post-launch plans.>
- <This material should have been presented at the PDR. Adopt the PDR slides and update them to capture updates in VVP v1r2.>



Validation of Operator Needs

- <Identify operator needs (operations and maintenance, or O&M) to be validated. The product or product component must be maintainable and supportable in its intended operational environment. Operator needs are typically found in the project's derived requirements, should be documented in the RAD, and should have been discussed in Section 4 ("Requirements") of this presentation. Most operator needs will be generic. Note which needs are specific to the project.>
- <Identify the tools and training available for O&M (e.g. Operations Manual, System Maintenance Manual, Process Assets)>
- <For each operator need, describe the scope of the validation. Usually, this will consist of simulations in the operational environment by the intended O&M personnel with the actual O&M tools and training in place.>
- <This material should have been presented at the PDR. Adopt the PDR slides and update them to capture updates in VVP v1r2.>



Validation of User Needs

- <Identify user needs (training, support, use of products) to be validated. These are typically found in the project's derived requirements, should be documented in the RAD, and should have been discussed in Section 4 ("Requirements") of this presentation. Many user needs will be generic. Note which needs are specific to the project.>
- <Identify the tools, training, and support services available to the user (e.g. External Users Manual) and the procedure for delivering these to the intended users>
- <For each user need, describe the scope of the validation. Usually, this will consist of simulations in a user environment by the intended users and/or beta testers with the actual User tools and training in place.>
- <This material should have been presented at the PDR. Adopt the PDR slides and update them to capture updates in VVP v1r2.>



- INTRODUCTION
- PDR REPORT
- OPERATIONS CONCEPT
- REQUIREMENTS
- ALGORITHM THEORETICAL BASIS
- SOFTWARE ARCHITECTURE AND INTERFACES
- DETAILED DESIGN DESCRIPTION
- QUALITY ASSURANCE
- **REQUIREMENTS ALLOCATION**
- RISKS AND ACTIONS
- SUMMARY AND CONCLUSIONS

Section 9 Setup Slide



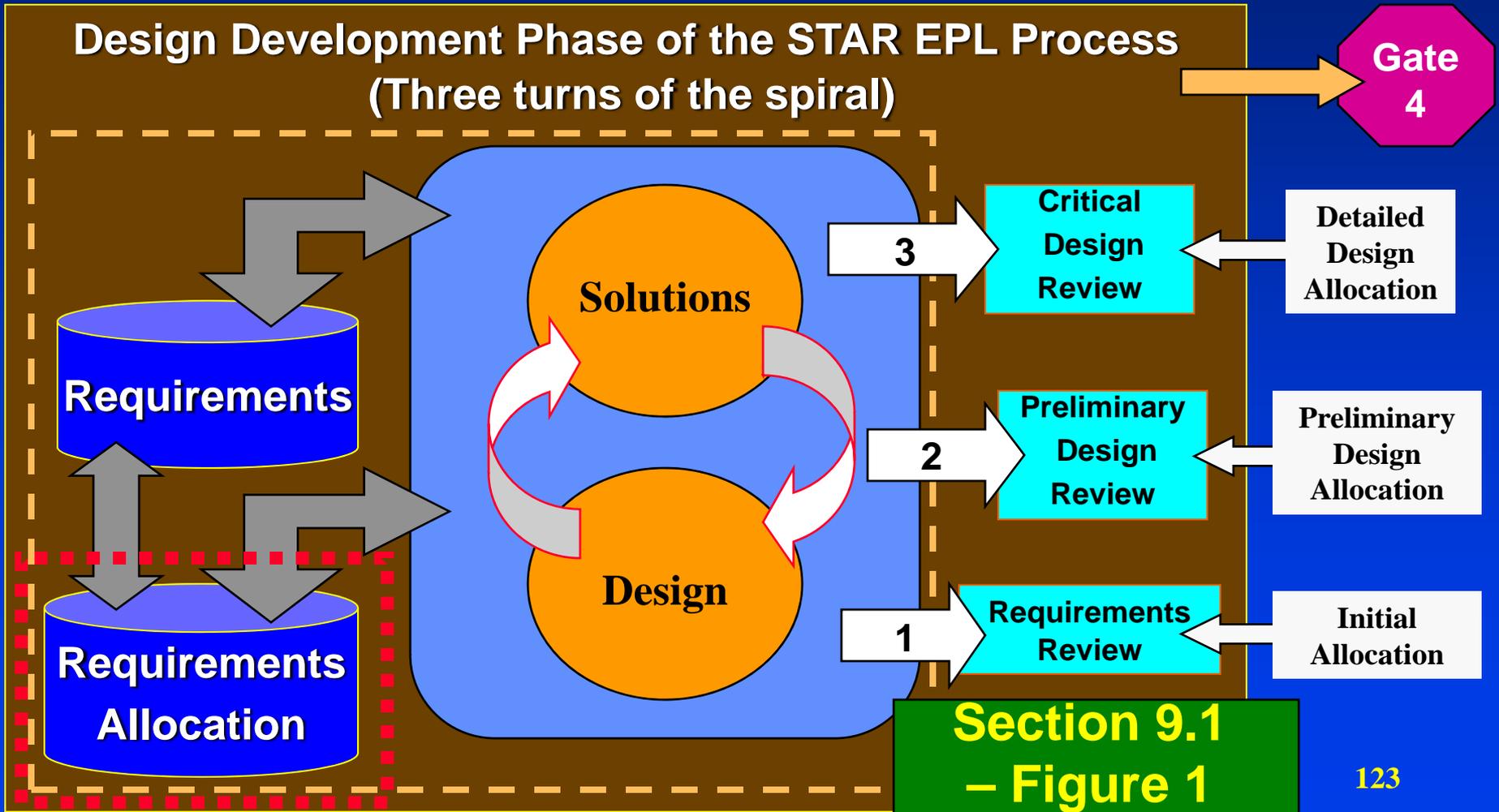
Section 9 - Requirements Allocation

Presented by

<Presenter's Name>
<Presenter's Title/Role>
<Presenter's Organization>



Iterative (Spiral) Development of Requirements Allocation





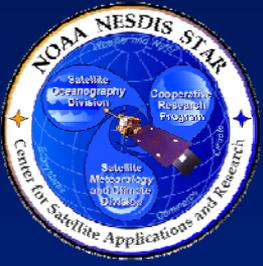
Requirements Allocation – Changes Since PDR

- **<List each requirements allocation change>**
 - » <If a derived requirement, list higher-level driving requirements>
 - » <If a basic requirement, list derived requirements that are affected>
 - » <Note whether the change is due to a new requirement, a changed requirement, or a design change>
 - If due to a new or changed requirement, specify the requirement and trace it to the requirements presentation in this CDD
 - If due to a design change, specify the change and trace it to the design presentation in this CDD
 - » <Note whether the change has been approved at a delta Requirements Review>
 - » <If the change has not been approved:>
 - <Explain rationale for the change (e.g., revealed by detailed design issue, operational constraint)>
 - <Note potential effects on the project plan>
 - <Document the agreement of affected stakeholders>
 - <Note new or modified risks that result from the change, to be summarized in Section 10 of this CDD>
 - <Note any recommended actions that result from the change, to be summarized in Section 10 of this CDD >



- INTRODUCTION
- PDR REPORT
- OPERATIONS CONCEPT
- REQUIREMENTS
- ALGORITHM THEORETICAL BASIS
- SOFTWARE ARCHITECTURE AND INTERFACES
- DETAILED DESIGN DESCRIPTION
- QUALITY ASSURANCE
- REQUIREMENTS ALLOCATION
- **RISKS AND ACTIONS**
- SUMMARY AND CONCLUSIONS

Section 10 Setup Slide



Section 10 – Risks and Actions

Presented by

<Presenter's Name>
<Presenter's Title/Role>
<Presenter's Organization>



<Project Name> - Risks at CDR

- There are <fill in the correct number> risks to be reviewed at the CDR
 - » <fill in the correct number> risks were identified at the PDR and documented in the PDRR
 - » <fill in the correct number> risks were identified after the PDR
- The following slides contain, for each risk item:
 - » A risk statement
 - » Risk assessment (Severity and Likelihood)
 - » Risk mitigation recommendation
 - » Status of actions identified to mitigate the risk



Risks from the PDR - Risk # 1

Raytheon

- **RISK # 1 - <Risk statement>**
- **Risk Assessment: <TBS>** (Severity = <TBS>, Likelihood = <TBS>).
<TBS = HIGH, MEDIUM, or LOW>
- **Risk Mitigation:** <Describe the risk mitigation plan, as stated in the PDR report. Use sub-bullets as warranted for clarity. Note actions associated with each item (sub-bullet) of the plan.>
- **Status:** <Present the development team's current assessment of the risk (HIGH, MEDIUM, LOW, or NONE). Explain the rationale for the assessment (e.g. list actions that are completed).>
- <Present status of actions associated with Risk # 1 in subsequent slides. Present completed actions, then open actions. Use separate slides for each action (see next 2 slides).>



Completed Actions - <Action number>

- ACTION: <Number> - <Action statement>
- CLOSURE CRITERIA: <Closure criteria statement>
- STATUS: Completed. <Demonstrate that the closure criteria have been met. Use multiple slides as necessary.>
- <Repeat for each completed action associated with Risk # 1>



Open Actions - <Action number>

- ACTION: <Number> - <Action statement>
- CLOSURE CRITERIA: <Closure criteria statement>
- CLOSURE PLAN: <Closure plan>
- STATUS: Open. <Explain what parts of the closure plan have been completed and what remains to be done. Note any additional risk to budget or schedule. Use multiple slides as necessary.>
- <Repeat for each open action associated with Risk # 1>



Risks from the PDR - Risk # 2

- <Present Risk # 2 status, using the same format as for Risk # 1>
- <On separate slides, present status of all actions associated with Risk # 2. Present completed actions, then open actions. Use the same format as for Risk # 1 actions.>
- <Repeat for each risk>
- <Then, present any new risks identified after the PRR (see next slide)>



New Risks - Risk # <N>

- **RISK # <N>** - <Risk statement>
- **Risk Assessment:** <TBS> (Severity = <TBS>, Likelihood = <TBS>). <TBS = HIGH, MEDIUM, or LOW>
- Risk Mitigation: <Describe the risk mitigation plan. Use sub-bullets as warranted for clarity. Note actions associated with each item (sub-bullet) of the plan.>
- <Present status of actions associated with Risk # N in subsequent slides. Present completed actions, then open actions. Use separate slides for each action (see next 2 slides).>



Completed Actions - <Action number>

- ACTION: <Number> - <Action statement>
- CLOSURE CRITERIA: <Closure criteria statement>
- STATUS: Completed. <Demonstrate that the closure criteria have been met. Use multiple slides as necessary.>
- <Repeat for each completed action associated with Risk # N>



Open Actions - <Action number>

- ACTION: <Number> - <Action statement>
- CLOSURE CRITERIA: <Closure criteria statement>
- CLOSURE PLAN: <Closure plan>
- STATUS: Open. <Explain what parts of the closure plan have been completed and what remains to be done. Note any additional risk to budget or schedule. Use multiple slides as necessary.>
- <Repeat for each open action associated with Risk # N>



New Risks - Risk # <N + 1>

- **RISK # <N + 1> - <Risk statement>**
- **Risk Assessment: <TBS>** (Severity = <TBS>, Likelihood = <TBS>). <TBS = HIGH, MEDIUM, or LOW>
- Risk Mitigation: <Describe the risk mitigation plan. Use sub-bullets as warranted for clarity. Note actions associated with each item (sub-bullet) of the plan.>
- <Present status of actions associated with Risk # N + 1 in subsequent slides, following the same format used for the Risk # N actions. Present completed actions, then open actions. Use separate slides for each action.>



Risk Summary - <N> Risks Can Be Closed

- <Present a bulleted list of risk statements for the risks that can be closed>
 - » <For each risk, list the associated actions that can be closed. Each of these should have been presented in Sections 9.1 or 9.2 as a completed action.>
 - » <Use multiple slides as necessary for clarity>



Risk Summary - <N> Risks Remain Open

- <Present a bulleted list of risk statements for the risks that are still open. Present HIGH risks first, followed by MEDIUM risks, then LOW risks.>
 - » <For each risk, list the actions that must be closed to reduce the risk to an acceptable level, with closure plans and estimated closure dates>



- INTRODUCTION
- PDR REPORT
- OPERATIONS CONCEPT
- REQUIREMENTS
- ALGORITHM THEORETICAL BASIS
- SOFTWARE ARCHITECTURE AND INTERFACES
- DETAILED DESIGN DESCRIPTION
- QUALITY ASSURANCE
- REQUIREMENTS ALLOCATION
- RISKS AND ACTIONS
- **SUMMARY AND CONCLUSIONS**

Section 11 Setup Slide



Section 11 - Summary and Conclusions

Presented by

<Presenter's Name>
<Presenter's Title/Role>
<Presenter's Organization>



Review Objectives Have Been Addressed

- <Explain how each review objective has been addressed>
- Preliminary Design Review Report and actions have been reviewed
 - » <Notable conclusions from this section>
 - Operations concept has been reviewed
 - » <Notable conclusions from this section>
 - Requirements changes have been reviewed
 - » <Notable conclusions from this section>
 - Algorithm theoretical basis has been reviewed
 - » <Notable conclusions from this section>
 - Software system architecture and interfaces have been reviewed
 - » <Notable conclusions from this section>



Review Objectives Have Been Addressed

- <Explain how each review objective has been addressed>
- Detailed design has been reviewed
 - » <Notable conclusions from this section>
- Plans for quality assurance have been reviewed
 - » <Notable conclusions from this section>
- Requirements allocation has been reviewed
 - » <Notable conclusions from this section>
- Risks and Actions have been reviewed
 - » <Notable conclusions from this section>



Issues, Actions And Risks

<List important issues, actions and risks that require attention. Use multiple slides as necessary for clarity.>

- <Item 1>
 - » <Conclusions about item 1>
-
 - »
- <Item N>
 - » <Conclusions about item N>



Next Steps

<List recommendations for next steps after the CDR>

- Preparation for Gate 4 Review
 - » <Recommendations for open actions>
 - » <Preparation of Gate 4 Review artifacts>
- Build phase
 - » <Recommendations for Build phase. Focus on preparation for Test Readiness Review.>



Open Discussion

- The review is now open for free discussion