# NOAA NESDIS
# CENTER for SATELLITE APPLICATIONS and RESEARCH (STAR)

# TASK GUIDELINE

## TG-9
## CODE & TEST DATA DEVELOPMENT (STEP 9) TASK GUIDELINES

## Version 3.0

TITLE: TG-9: CODE & TEST DATA DEVELOPMENT (STEP 9) TASK GUIDELINE VERSION 3.0

AUTHORS:

Ken Jensen (Raytheon Information Solutions)

VERSION HISTORY SUMMARY

| Version | Description | Revised Sections | Date |
|---|---|---|---|
| 1.0 | No version 1 | | |
| 2.0 | No version 2 | | |
| 3.0 | New Task Guideline adapted from CMMI guidelines by Ken Jensen (Raytheon Information Solutions) | New Document | 10/01/2009 |
| | | | |

## TABLE OF CONTENTS

Page

## LIST OF FIGURES

Page

## LIST OF TABLES

Page

## LIST OF ACRONYMS

| | |
|---|---|
| ATBD | Algorithm Theoretical Basis Document |
| BB | Baseline Build |
| CDD | Critical Design Document |
| CDR | Critical Design Review |
| CDRR | Critical Design Review Report |
| CI | Cooperative Institute |
| CICS | Cooperative Institute for Climate Studies |
| CIMSS | Cooperative Institute for Meteorological Satellite Studies |
| CIOSS | Cooperative Institute for Oceanographic Satellite Studies |
| CIRA | Cooperative Institute for Research in the Atmosphere |
| CL | Check List |
| CLI | Check List Item |
| CoRP | Cooperative Research Program |
| CM | Configuration Management |
| CMMI | Capability Maturity Model Integration |
| CREST | Cooperative Remote Sensing and Technology Center |
| CTR | Code Test Review |
| DDD | Detailed Design Document |
| DG | Document Guidelines |
| DM | Data Management |
| DPP | Development Project Plan |
| DPR | Development Project Report |
| EPG | Enterprise Process Group |
| EPL | Enterprise Product Lifecycle |
| IPT | Integrated Product Team |
| NESDIS | National Environmental Satellite, Data, and Information Service |
| NOAA | National Oceanic and Atmospheric Administration |
| OCD | Operations Concept Document |

| | |
|---|---|
| PAR | Process Asset Repository |
| PBR | Project Baseline Report |
| PDD | Preliminary Design Document |
| PDR | Preliminary Design Review |
| PDRR | Preliminary Design Review Report |
| PG | Process Guidelines |
| PP | Project Proposal |
| PRD | Project Requirements Document |
| PRG | Peer Review Guidelines |
| PRRR | Project Requirements Review Report |
| PSR | Project Status Report |
| QA | Quality Assurance |
| R&D | Research & Development |
| RAD | Requirements Allocation Document |
| RAS | Requirements Allocation Sheet |
| RNM | Requirements/Needs Matrix |
| SG | Stakeholder Guideline |
| SPSRB | Satellite Products and Services Review Board |
| STAR | Center for Satellite Applications and Research |
| STP | System Test Plan |
| SWA | Software Architecture Document |
| TD | Training Document |
| TG | Task Guideline |
| TRD | Test Readiness Document |
| TRR | Test Readiness Review |
| TRRR | Test Readiness Review Report |
| UTP | Unit Test Plan |
| VVP | Verification and Validation Plan |

## 1. INTRODUCTION

The NOAA/NESDIS Center for Satellite Applications and Research (STAR) develops a diverse spectrum of complex, often interrelated, environmental algorithms and software systems. These systems are developed through extensive research programs, and transitioned from research to operations when a sufficient level of maturity and end-user acceptance is achieved. Progress is often iterative, with subsequent deliveries providing additional robustness and functionality. Development and deployment is distributed, involving STAR, the Cooperative Institutes (CICS[1], CIMSS[2], CIOSS[3], CIRA[4], CREST[5]) distributed throughout the US, multiple support contractors, and NESDIS Operations.

NESDIS/STAR is implementing an increased level of process maturity to support the development of these software systems from research to operations. This document is a Task Guideline (TG) for users of this process, which has been designated as the STAR Enterprise Product Lifecycle (EPL).

### 1.1. Objective

The STAR EPL is designed as a sequence of 11 process steps that take a product from initial conception through delivery to operations. These steps are:

- Step 1 - Basic Research (TG-1)
- Step 2 - Focused R & D (TG-2)
- Step 3 - Project Proposal (TG-3)
- Step 4 - Resource Identification (TG-4)
- Step 5 - Development Project Plan (TG-5)
- Step 6 - Project Requirements (TG-6)
- Step 7 - Preliminary Design (TG-7)
- Step 8 - Detailed Design (TG-8)

[1] Cooperative Institute for Climate Studies

[2] Cooperative Institute for Meteorological Satellite Studies

[3] Cooperative Institute for Oceanographic Satellite Studies

[4] Cooperative Institute for Research in the Atmosphere

[5] Cooperative Remote Sensing and Technology Center

- **Step 9 - Code & Test Data Development (TG-9)**
- Step 10 - Code Test And Refinement (TG-10)
- Step 11 - System Integration and Test (TG-11)

The objective of this Task Guideline (TG-9) is to describe how to perform the standard tasks of STAR EPL process step 9, “Code & Test Data Development”.

The intended users of this TG are all participants in the STAR EPL process who are involved in performing the standard tasks of step 9. Participants are referred to as STAR EPL stakeholders.

To determine whether or not they should be involved with this step, the readers of this TG should first determine what stakeholder roles apply to their participation in a STAR research-to-operations development project. Generic stakeholder roles are listed in Section 3 of this TG and discussed in Section 3.2 of the EPL Process Guideline (PG-1)[6]. PG-1 and this TG will direct stakeholders to Stakeholder Guidelines (SG) that are pertinent to their roles.

### 1.2. Version History

This is the first version of TG-9. It is identified as version 3.0 to align it with the release of the version 3.0 STAR EPL process assets.

### 1.3. Overview

This TG contains the following sections:

Section 1.0 - Introduction
Section 2.0 - References
Section 3.0 - Stakeholders
Section 4.0 - Test Readiness Review
Section 5.0 - Project Artifacts
Section 6.0 - Task Descriptions

[6] It is recommended that potential STAR EPL stakeholders either review PG-1 prior to using this TG or use it as a reference while using this TG.

## 2. REFERENCE DOCUMENTS

All of the reference documents for the STAR EPL process are STAR EPL process assets that are accessible in a Process Asset Repository (PAR) on the STAR website. http://www.star.nesdis.noaa.gov/star/EPL_index.php.

Process assets include:

- Process Guidelines
- Stakeholder Guidelines
- Task Guidelines
- Peer Review Guidelines
- Review Check Lists
- Document Guidelines
- Training Documents

### 2.1. Process Guidelines

Process Guideline (PG) documents describe STAR's standard set of practices and guidelines for tailoring them to specific projects.

- STAR EPL Process Guidelines (PG-1)
- STAR EPL Process Guidelines Appendix (PG-1.A)
- STAR EPL Tailoring Guidelines (PG-2)

PG-1 and PG-1.A apply generally to each EPL step. Each stakeholder performing tasks during each step can benefit from a familiarity with these documents.

PG-2 is primarily useful for project planners and project plan reviewers during steps 4 and 5. It is also useful during steps 6-11 for project plan revision tasks.

### 2.2. Stakeholder Guidelines

A Stakeholder Guideline (SG) is a description of how to perform all STAR EPL standard tasks assigned to a given type of stakeholder. It should itemize the actions to be taken. It should contain appropriate standards, conventions, and (where appropriate) examples. It should point to the appropriate references and the required artifacts.

Stakeholder roles are identified in Section 3 of this TG. For each type of stakeholder, the appropriate SG provides that stakeholder with a complete description of the standard tasks for that stakeholder role, along with references to all appropriate process assets and project artifacts (c.f. Section 5 of this TG). This functions as a complement to the TGs (c.f. Section 2.3 of this TG), which provide a completion description of all stakeholder tasks for a specific process step.

Table 2.2.1 lists the Stakeholder Guidelines that are relevant to this step.

**TABLE 2.2.1 –** Stakeholder Guidelines for Step 9

| ID | Stakeholder |
|---|---|
| SG-4 | STAR CM/DM |
| SG-5 | STAR Web Developers |
| SG-6 | STAR Quality Assurance |
| SG-13 | Development Leads |
| SG-14 | Development Scientists |
| SG-15 | Development Testers |
| SG-16 | Development Programmers |
| SG-17 | Technical Review Leads |
| SG-18 | Technical Reviewers |

### 2.3. Task Guidelines

A Task Guideline (TG) is a description of how to perform the tasks of a STAR EPL process step. It should itemize the actions to be taken. It should contain appropriate standards, conventions, and (where appropriate) examples. It should point to the appropriate references and the required artifacts. There is one Task Guideline for each step in the STAR EPL. The relevant TG for this step is TG-9 (this document).

### 2.4. Peer Review Guidelines

For each review (c.f. Section 4), there is a Peer Review Guideline (PRG) that describes the objectives of the review, the required artifacts, standards for reviewers, requirements for approval, and options other than approval. For step 9, the relevant PRGs include:

- Test Readiness Review Guidelines (PRG-9)

### 2.5. Review Check Lists

For each review (c.f. Section 4), there is a Review Check List (CL) that captures all the objectives for a review as a set of check list items. Each item in the check list should have a "Disposition" column that contains "Pass", "Conditional Pass", "Defer", "Waive", or "N/A" (Not Applicable). Each item will also have columns for Risk Assessment and for Actions generated. For step 9, the relevant CLs include:

- Test Readiness Review Check List (CL-9)

### 2.6. Document Guidelines

There is a Document Guideline (DG) for each standard STAR EPL document. Each DG includes a description of the purpose for the document, a standard document outline (table of contents), a brief description of each subsection in the outline, and an Appendix containing an example document.

Table 2.6.1 lists the Document Guidelines that are relevant to this step.

**TABLE 2.6.1 –** Document Guidelines for Step 9

| ID | Document |
|---|---|
| DG-1.2 | Software Architecture Document (SWA) |
| DG-5.1 | Development Project Plan (DPP) |

| | |
|---|---|
| DG-5.4 | Project Baseline Report (PBR) |
| DG-6.2 | Requirements Allocation Document (RAD) |
| DG-6.3 | Verification and Validation Plan (VVP) |
| DG-8.1 | Detailed Design Document (DDD) |
| DG-9.1 | Unit Test Plan (UTP) |
| DG-9.2 | Test Readiness Document (TRD) |
| DG-9.2.A | TRD Appendix |
| DG-9.3 | Test Readiness Review Report (TRRR) |

### 2.7. Training Documents

Training Documents (TD) assist the stakeholders (c.f. Section 3) in performing the process tasks. By using the TDs, the stakeholders should be able to perform the tasks more effectively.

Table 2.7.1 lists the Training Documents that are relevant to this step.

**TABLE 2.7.1 –** Training Documents for Step 9

| ID | Training Document |
|---|---|
| TD-11.1 | FORTRAN Coding Standards |
| TD-11.1.A | Transition from Fortran 77 to Fortran 90 |
| TD-11.2 | C Coding Standards |

## 3. STAKEHOLDERS

The STAR Enterprise is comprised of a large number of organizations that participate and cooperate in the development and production of environmental satellite data products and services. Individual project teams are customarily composed of personnel from these organizations, supplemented by contractor personnel. These organizations and project teams are referred to as the STAR Enterprise stakeholders.

An overview of the stakeholder roles is provided in the STAR EPL Process Guidelines (PG-1, c.f. Section 2). A more detailed description can be found in the Stakeholder Guidelines (SGs, c.f. Section 2).

Stakeholders who have a role during step 9 include:

- STAR CM/DM (SG-4)
- STAR Web Developer (SG-5)
- STAR QA (SG-6)
- Development Lead (SG-13)
- Development Scientist (SG-14)
- Development Tester (SG-15)
- Development Programmer (SG-16)
- Technical Review Lead (SG-17)
- Technical Reviewer (SG-18)

***STAR CM/DM*** is the Configuration Management (CM) and Data Management (DM) group for the STAR organization. CM/DM is responsible for establishing and maintaining project baselines for code, test data, documentation, and reports. CM/DM works with each Development Lead to ensure that project artifacts are maintained in accordance with STAR standards. CM/DM works with Operations CM/DM on the transition of the project baseline from pre-operational development to operations.

***STAR Web Developer*** is responsible for maintenance of the STAR web pages. The Web Developer works with STAR CM/DM to ensure that all project baseline items are posted to the appropriate project artifact repository in a timely fashion. The Web Developer works with the STAR EPG and STAR CM/DM to ensure that all STAR EPL process assets are posted to the PAR, and to ensure that all process measures are posted to the STAR Measurement Repository.

***STAR QA*** is the quality assurance (QA) group for the STAR organization. QA is responsible for ensuring that each project's tailored process meets STAR EPL process standards and ensuring that each project meets its process requirements during its pre-operational development phases. QA works with the STAR EPG to ensure effective implementation of the process throughout the organization.

***Development Lead*** is nominally a STAR scientist who leads a project’s development efforts after a Project Proposal (PP) has been approved. The Development Lead is typically identified in the PP and is often the same person who was the Research Lead. The Development Lead works with STAR Management to tailor the STAR EPL process to the project and leads the project's development efforts during the Design and Build phases as the lead of the Integrated Product Team (IPT).

***Development Scientist*** is nominally a STAR scientist who has been assigned by the Development Lead to one or more of the tasks of reviewing the technical content of project proposals, maturing a research algorithm into an operational algorithm, developing project requirements, supporting product design, coding and testing, and providing product validation and science maintenance.

***Development Tester*** is any person located at a research organization who has been assigned by the Development Lead to one or more of the tasks of identifying pre-operational test data, acquiring and integrating the test data into the pre-operational product processing system, creating pre-operational unit and system test plans, executing unit and system tests, and analyzing and reporting test results for review.

***Development Programmer*** is a programmer who has been assigned by the Development Lead to one or more of the tasks of preliminary design and detailed design of pre-operational code, writing pre-operational code, integrating code into a pre-operational system, and supporting Development Testers in testing pre-operational code.

***Technical Review Lead*** is responsible for leading the team of Technical Reviewers for one or more of the six Technical Reviews. The Technical Review Lead works with the Development Lead and the Technical Reviewers to ensure that the review is prepared for, conducted, and closed according to review standards.

***Technical Reviewer*** is responsible for reviewing and approving project artifacts and project status at one or more of the six Technical Reviews. Technical Reviewers work with the Technical Review Lead to ensure that the review is prepared for, conducted, and closed according to review standards.

Stakeholder satisfaction is a critical component of the process. The intention is for the process to be more of a benefit that a burden to stakeholders. If stakeholders are not satisfied that this is the case, the process will require improvement.

Stakeholders are strongly encouraged to provide feedback to the EPG. Comments and suggestions for improvement of the process architecture, assets, artifacts and tools are always welcome. Stakeholders can provide feedback by contacting:

Ken.Jensen@noaa.gov

## 4. TEST READINESS REVIEW

Test Readiness Review (TRR) is a Build Phase Technical Review. Its purpose is to determine whether code and test data development are sufficient to allow testing of the pre-operational software components (unit testing). Upon successful completion of this review, step 10 (Code Test and Refinement) commences.

Standard TRR objectives:

- Identify relevant stakeholders and document their involvement according to the project plan.
- Review the software architecture, including external interfaces, and identify changes since CDR.
- Identify changes to the detailed design since CDR.
- Identify changes to the verification and validation plan since CDR.
- Demonstrate the test readiness of each unit in the software architecture.
- Provide all applicable technical data to support unit testing, including:
  - Pre-operational code and test data
  - Unit test plan
- Identify and evaluate risks. Recommend risk mitigation activities.
- Document the closing of all action items since CDR. Make recommendations for open actions and new actions.

Standard TRR entry criteria:

- Entry # 1 - A Critical Design Review Report (CDRR) has been written. The TRR reviewers have access to the current baseline version of the CDRR.
- Entry # 2 - A Development Project Plan (DPP) has been written. The TRR reviewers have access to the current baseline version of the DPP.
- Entry # 3 - A Requirements Allocation Document (RAD) has been written. The TRR reviewers have access to the current baseline version of the RAD.

- Entry # 4 - A Software Architecture Document (SWA) has been written. The TRR reviewers have access to the current baseline version of the SWA.
- Entry # 5 - A Detailed Design Document (DDD) has been written for each software unit in the software architecture. The TRR reviewers have access to the current baseline version of the DDDs.
- Entry # 6 - A Verification and Validation Plan (VVP) has been written. The TRR reviewers have access to the current baseline version of the VVP.
- Entry # 7 - A Unit Test Plan (UTP v1r0) has been written. The TRR reviewers have access to the current baseline version of the UTP.
- Entry # 8 – Pre-operational code to implement the detailed design is accessible to the TRR reviewers.
- Entry # 9 - Pre-operational test data, including "truth" data is accessible to the TRR reviewers.
- Entry # 10 - A Project Baseline Report (PBR v2r0) has been written. The TRR reviewers have access to the current baseline version of the PBR.
- Entry # 11 - A Test Readiness Document (TRD) has been written. The TRR reviewers have access to the current baseline version of the TRD.

Standard TRR exit criteria:

- Exit # 1 - CDR "Conditional Pass" items have been satisfactorily disposed of.
- Exit # 2 - CDR "Defer" items have been satisfactorily disposed of.
- Exit # 3 - Changes to the project plan since Gate 4 Review are approved.
- Exit # 4 - Requirements allocation changes since CDR are approved.
- Exit # 5 - Changes to external interfaces since CDR are approved.
- Exit # 6 - Changes to the software architecture since CDR are approved.
- Exit # 7 - Changes to the detailed design since CDR are approved.
- Exit # 8 - Changes to the verification and validation plan since CDR are approved.
- Exit # 9 - The unit test plan and UTP are satisfactory
- Exit # 10 - Pre-operational code to implement the detailed design has been written according to standards and has been built into executable units.

- Exit # 11 - Pre-operational test data, including "truth" data, are satisfactory.
- Exit # 12 - The project baseline and PBR are satisfactory.
- Exit # 13 - The project artifacts document all approved changes to requirements, requirements allocation, external interfaces, software architecture, detailed design, and verification and validation plan since the CDR.
- Exit # 14 - The TRRR documents the current status of project risks, actions and TRR exit criteria.
- Exit # 15 - Project risks and actions are acceptable. Project is ready for unit testing.

Refer to PRG-9 for a more detailed description of the TRR. The standard TRR entry criteria, exit criteria, and check list is documented in the process asset CL-9 (c.f. Section 2).

TRR objectives, entry criteria, exit criteria, and check list may be tailored. Tailoring guidelines are provided in the process asset PG-2 (c.f. Section 2). Refer to the Development Project Plan (DPP) Section 5 to determine whether there has been any project-specific tailoring for the TRR.

## 5. PROJECT ARTIFACTS

Project Artifacts are a set of items that must be produced by the appropriate stakeholders during the product life cycle to support the reviews. They are established and maintained under Configuration Management (CM) by an Enterprise Process Group (EPG) under the direction of a Steering Committee.

The project artifacts are maintained in a project artifact repository. This is a complete set of configuration-managed artifacts developed by each project in accordance with STAR standards. When a project artifact has been approved at a Technical Review or Gate Review, it is placed in the project artifact repository under CM.

Project artifacts that are recommended for development during step 9 are listed in Table 5.1.

**TABLE 5.1 –** Step 9 Artifacts

| Artifact | Type | Review | Baseline Build |
|---|---|---|---|
| Development Project Plan v3.0 | Document | TRR | 3.0 |
| Pre-Operational Code | Code | TRR | 3.0 |
| Pre-Operational Test Data | Test Data | TRR | 3.0 |
| Software Architecture Document v2.2 | Document | TRR | 3.0 |
| Detailed Design Document v1.1 | Document | TRR | 3.0 |
| Requirements Allocation Document v1.3 | Document | TRR | 3.0 |
| Unit Test Plan v1.0 | Document | TRR | 3.0 |
| Verification and Validation Plan v1.3 | Document | TRR | 3.0 |
| Test Readiness Document | Presentation | TRR | 3.0 |
| Project Baseline Report v3.0 | Report | TRR | 3.0 |
| Test Readiness Review Report | Report | CTR | 3.1 |
| Project Baseline Report v3.1 | Report | None | 3.1 |

***Development Project Plan v3.0:*** The Development Project Plan (DPP) documents the plan for the development, testing, review, and transition to operations for the project, including stakeholders, tasks, work breakdown structure (WBS), schedule and resources. DPP v2 should have been produced in step 8 for the Gate 4 Review. DPP v3.0 is a planned revision that should respond to Gate 4 Review actions and adjustments that may be needed during code and test data development. Refer to DG-5.1 for detailed DPP guidelines.

***Pre-Operational Code:*** The Pre-Operational Code (PCOD v1.x) consists of all software components of the detailed design that was approved at the CDR, ready for unit testing. It is expected that SPSRB coding standards will be applied to the pre-operational code. Currently, coding standards exist for FORTRAN, C, and C++ code, and general programming standards exist for all code. These standards are found on the SPSRB web site at http://projects.osd.noaa.gov/spsrb/standards_prog.htm. This requirement may be waived if the circumstances of a specific project provide a compelling reason for a waiver. Waivers should be agreed to as early as possible, included in the project plan, and accepted by operations prior to unit testing.

***Pre-Operational Test Data:*** Pre-Operational Test Data (PTEST v1.x) are the data files used for unit testing of the Pre-Operational Code, including the input data and output data identified in the current baseline versions of the Algorithm Theoretical Basis Document (ATBD) and Software Architecture Document (SWA).

***Software Architecture Document v2.2:*** The SWA provides the software architecture for the processing code that will implement the algorithm. SWA v2.1, produced for the CDR, should have described the software architecture as it existed at the completion of the Design Phase of the STAR EPL. Often, the SWA must be revised (v2.2) to document revisions to the software architecture that occur during pre-operational code development. Refer to DG-1.2 for detailed SWA guidelines.

***Detailed Design Document v1.1:*** The purpose of the Detailed Design Document (DDD) is to describe the product design at a level of detail that is sufficient for the development programmers to write fully functional pre-operational code. A separate DDD is produced for each software unit that is part of the product processing system. The software units are the Layer-2 elements that are defined in the system layer product software architecture, as described in the SWA. DDD v1.0, produced for the CDR, should have described the detailed design as it exists at the completion of the Design Phase of the STAR EPL. Often, the DDD must be revised (v1.1) to document revisions to the detailed design that occur during pre-operational code development. Refer to DG-8.1 for detailed DDD guidelines.

***Requirements Allocation Document v1.3:*** The Requirements Allocation Document (RAD) contains the basic and derived requirements for the work products and the allocation of the requirements to system components and product components. RAD v1.2, produced for the CDR, should have documented the requirements and requirements allocation at the end of the Design phase of the STAR EPL. Often, the RAD must be revised (v1.3) to document revisions to the requirements allocation that are needed to reflect detailed design changes that occur during pre-operational code development. Refer to DG-6.2 for detailed RAD guidelines.

***Unit Test Plan v1.0:*** The Unit Test Plan (UTP) contains the test plan for each software unit in the project's product processing system. The UTP, a complement to the project's Verification and Validation Plan (VVP), focuses on the specifics of the software units and the testing of their functionality and performance. The UTP should document the purpose and function of each unit, its traceability to the project requirements, unit data flows, unit components and unit functions to be tested, a test data description, planned test methods and test sequences and identified test risks. Refer to DG-9.1 for detailed UTP guidelines.

***Verification and Validation Plan v1.3:*** The Verification and Validation Plan (VVP) describes the work products to be verified and validated, the requirements for each selected work product and the verification and validation methods for each selected work product. VVP v1.3, produced for the TRR, includes any revisions needed to maintain consistency with the UTP and the Requirements Allocation Document (RAD). Refer to DG-6.3 for detailed VVP guidelines.

***Test Readiness Document:*** The Test Readiness Document (TRD) consists of the presentation slides for the Test Readiness Review (TRR). Refer to DG-9.2 and DG-9.2.A for detailed TRD guidelines.

***Project Baseline Report v3.0:*** The Project Baseline Report (PBR v3.0) is the document that describes the status of the configuration items that comprise the project baseline at the TRR. Refer to DG-5.4 for detailed PBR guidelines.

Note that these artifacts are typically included in STAR Baseline Build (BB) 3.0. BB 3.0 provides the artifacts for the TRR. **STAR CM/DM** executes BB 3.0, in consultation with the developers of the BB 3.0 artifacts.

***Test Readiness Review Report:*** The TRR Report (TRRR) summarizes the TRR Reviewers' assessment of the project readiness for unit testing, including identified risks and risk mitigation actions. Refer to DG-9.3 for detailed TRRR guidelines.

***Project Baseline Report v3.1:*** When the TRRR is completed, it is added to the baseline for BB 3.1. The PBR is updated to v3.1 to include the addition of the TRRR as well as any TRR artifacts that are revised as the result of TRR actions.

## 6. TASK DESCRIPTION

### 6.1 Pre-Operational System Development Process

Pre-operational system development is an iterative process that occurs throughout the Build phase of the product lifecycle. This phase includes three steps that produce an integrated product processing system through an iterative (spiral) development of code, test data and test plans.

- **Code & Test Data Development (step 9 of the STAR EPL)**
- Code Test & Refinement (step 10 of the STAR EPL)
- System Integration & Test (step 11 of the STAR EPL)

Figure 6.1 illustrates the pre-operational system development process, with step 9 highlighted.

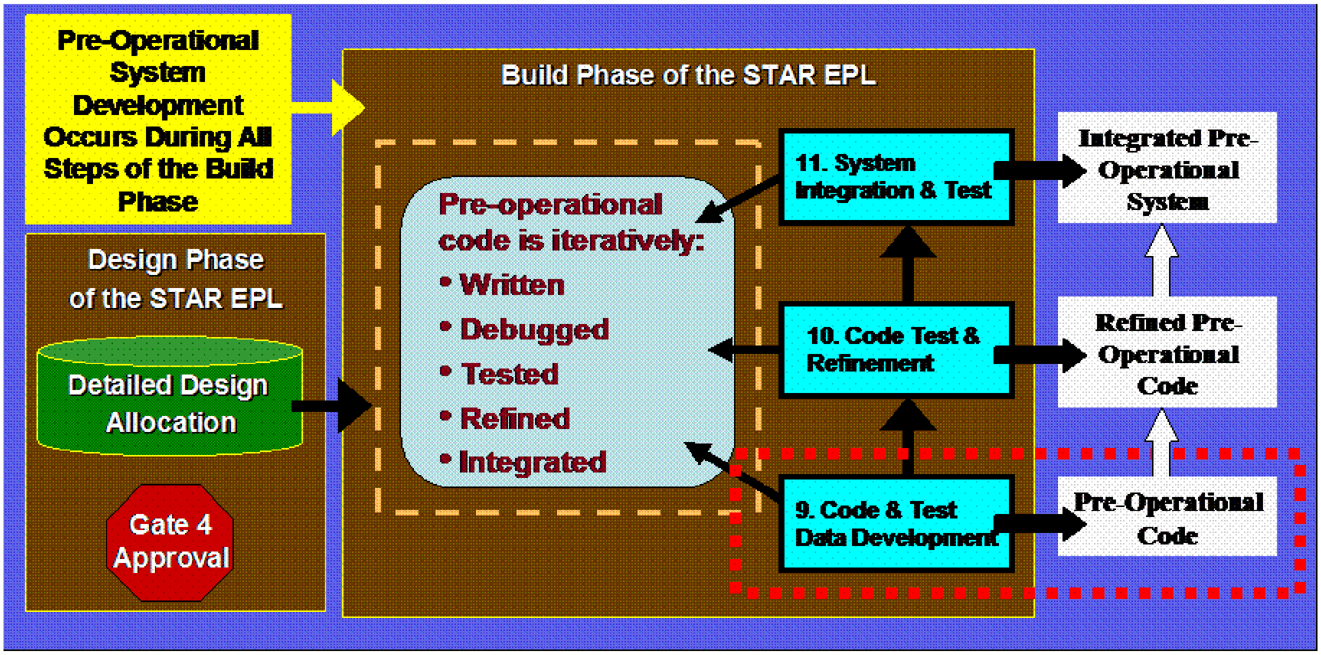
Pre-Operational System Development Occurs During All Steps of the Build Phase
Design Phase of the STAR EPL
Detailed Design Allocation
Gate 4 Approval
Build Phase of the STAR EPL
Pre-operational code is iteratively:
• Written
• Debugged
• Tested
• Refined
• Integrated
11. System Integration & Test
10. Code Test & Refinement
9. Code & Test Data Development
Integrated Pre-Operational System
Refined Pre-Operational Code
Pre-Operational Code

**Figure 6.1** – Pre-Operational System Development Process

As Figure 6.1 shows, the objective of step 9 is to produce pre-operational code. Pre-operational code consists of the system components in the detailed design (software units and sub-units). This code will be written and debugged, and ready for unit testing, but not formally tested.

The process of producing a complete pre-operational product processing system involves writing, debugging, testing, refining, and integrating the code. Because these functions affect each other, the process is inherently iterative. Figure 6.2 illustrates this.

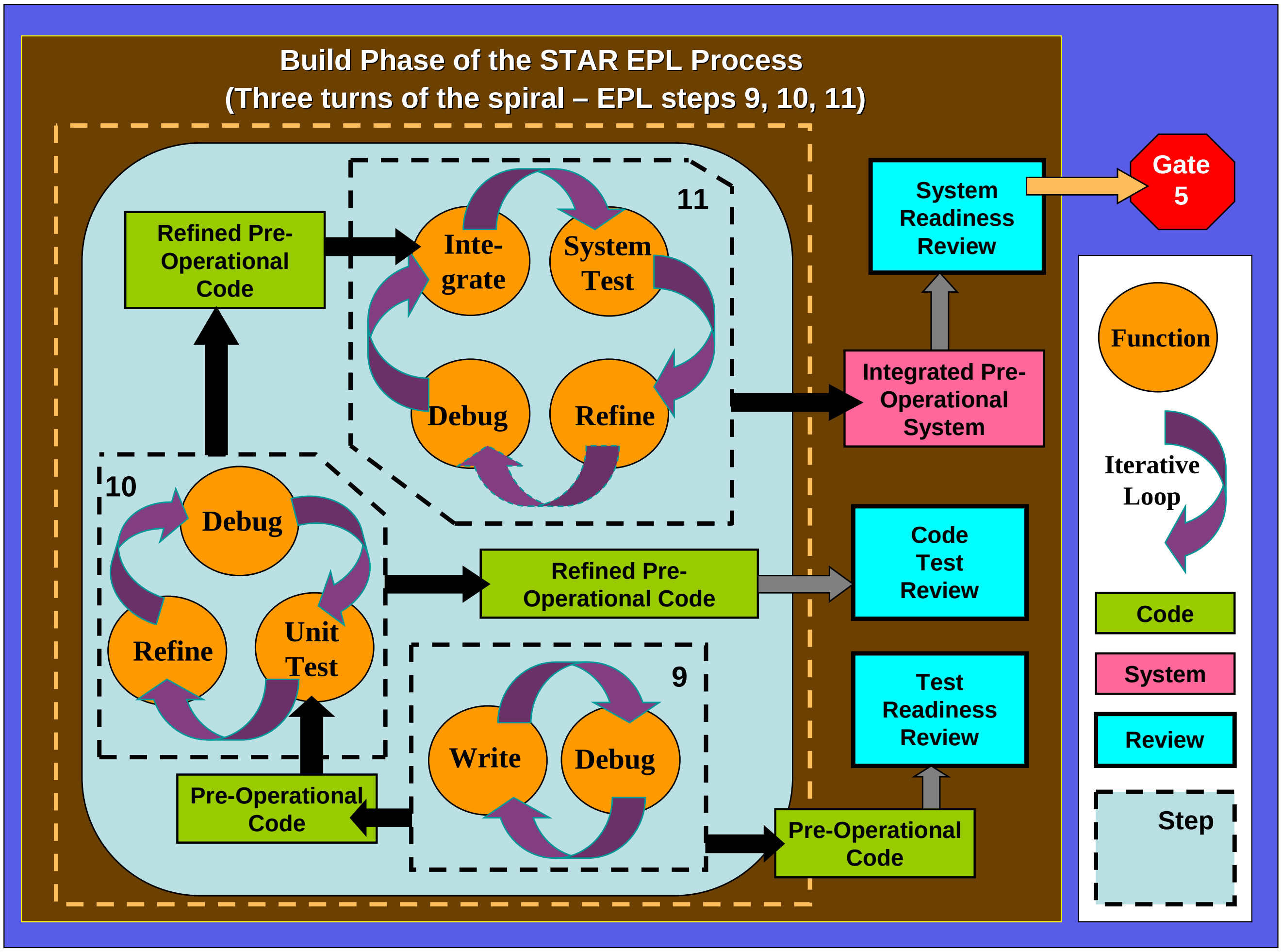

**Figure 6.2** – Iterative (Spiral) Development of Pre-Operational System

Note that steps 10 and 11 continue pre-operational system development. The details of these steps are described in TG-10 and TG-11 respectively, but it is important to note that refinement of the step 9 pre-operational code and integration into a complete pre-operational product processing system is expected during these steps. Therefore, the objective of the step 9 activities (write and debug code) is limited to producing code that is sufficiently mature and complete to allow unit testing. This is illustrated in Figure 6.2 as the output from step 9 is pre-operational code that is input to the Unit Test function of step 10.

## 6.2 Code & Test Data Development Process Flow

Figure 6.3 shows the process flow for step 9.

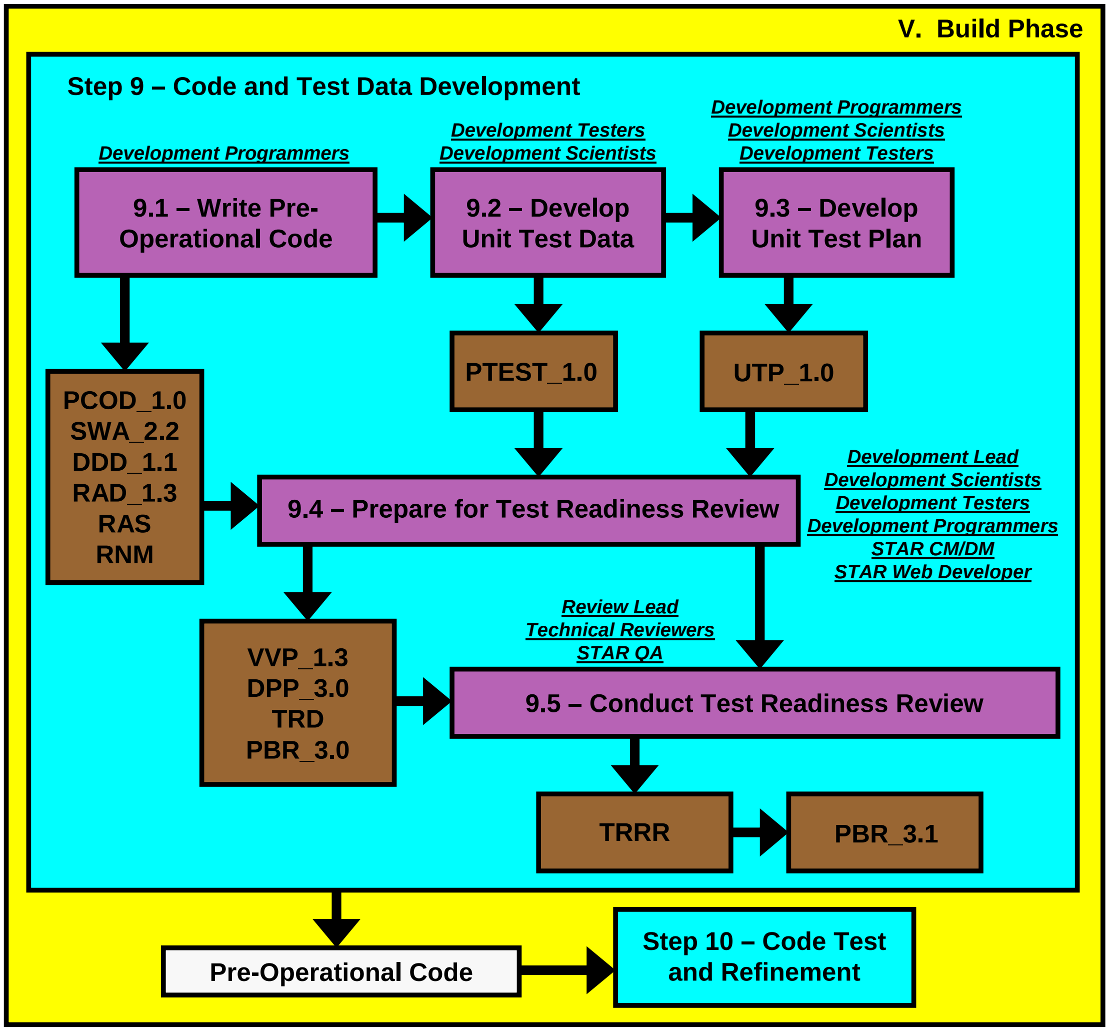

**Figure 6.3** – Step 9 Process Flow

### 6.3 Expected BEGIN State

- REQUIRED: A  CDR has been conducted
- REQUIRED: A  Gate 4 Review has been conducted
- REQUIRED: A Detailed Design Allocation has been developed and approved
- REQUIRED: Baseline Build (BB) 2.6 has placed the following items in the project artifact repository:
    - DPP, including Appendices
    - RAD, including Appendices
    - VVP
    - SWA
    - DDD
    - Critical Design Document (CDD)
    - Critical Design Review Report (CDRR)
    - Gate 4 Document (G4D)
    - Gate 4 Review Report (G4RR)
    - PSR, including Appendix
    - PBR
- EXPECTED: BB 2.6 has placed the following items in the project artifact repository:
    - R&D code
    - R&D test data
    - Project Proposal (PP)
    - Gate 2 Review Report (G2RR)
    - Gate 3 Review Report (G3RR)
    - Operations Concept Document (OCD)
    - ATBD
    - Project Requirements Document (PRD)
    - Project Requirements Review Report (PRRR)
    - Preliminary Design Document (PDD)
    - Preliminary Design Review Report (PDRR)
- REQUIRED: PBR_2.6 documents the status of the BB 2.6 project baseline

- REQUIRED: Gate 4 reviewers have approved the project to proceed to the Code and Test Data Development step, and have documented this approval in the G4RR.

### 6.3.1 Task Inputs

Task inputs consist of the following BB 2.6 items:

- DPP_2.0
- SWA_2.1
- RAD_1.2
- Requirements/Needs Matrix (RNM)
- Requirements Allocation Sheet (RAS)
- VVP_1.2
- DDD_1.0
- CDD
- CDRR
- PSR_2.0, including Appendix
- G4D
- G4RR
- PBR_2.6

### 6.3.2 Corrective Actions

The CDRR and G4RR will document any actions that are needed to reduce risk during step 9. Usually, these actions should be closed before the TRR.

Additional corrective actions are typically generated during step 9, to mitigate project risks that are identified during code and test data development. Project risks and risk mitigation actions should be identified in the PSR Appendix.

The needed corrective actions may require revisions to the project plan, typically by the addition of sub-tasks and revisions to the task schedule. The **Development Lead** should determine whether these revisions are manageable or are so significant that a re-plan is needed. If necessary, the **Development Lead** should consult STAR Management on the advisability of a re-plan. Re-planning is expected to be a rare event, but it may occur if the project requirements have added significant scope or if unexpected technical issues have been discovered.

#### 6.3.2.1 Delta Gate 4 Review

If it is determined that a re-plan is needed, the **Development Lead** should consult STAR Management to determine whether there should be a delta Gate 4 Review. This determination should depend upon whether the re-plan will significantly affect the pre-operational code. If so, a delta Gate 4 Review should be prepared for and conducted in the same manner as the normal Gate 4 Review. Refer to the step 8 Task Guideline (TG-8) and the Gate 4 Peer Review Guideline (PRG-8.2) for guidance. Following approval of the re-plan, the project can return to its step 9 activities under the new plan.

### 6.4 Desired END State

- A Detailed Design Allocation of the requirements identifies product and system components down to the Sub-Unit-Layer, and traces each component to one or more requirement.
- A plan has been developed for monitoring the status of the requirements and their allocation to ensure that the integrity of the requirements allocation is preserved as the implementation of the detailed design proceeds through the Build phase.
- The functionality of all system components in the detailed design (software units and sub-units) has been implemented in pre-operational code that meets coding standards.
- Pre-operational code has been debugged, compiled, and built into executable software units, ready for unit testing in the designated test environment.
- A plan for unit testing has been developed. The plan ensures that the unit tests will address all required code functionality and code outputs.
- All data required for implementation of the unit test plan has been acquired or developed, and is available in the designated test environment.
- The project plan has been updated as necessary
- The status of project risks and actions has been updated

- A TRR of the project plan, software architecture, and unit test readiness has been conducted
- A TRRR has been written. The TRRR approves the project to proceed to the next step, Code Test and Refinement.
- Baseline Build 3.1 has placed the required items in the project artifact repository
- PBR_3.1 documents the status of the BB 3.1 project baseline

### 6.4.1 Task Outputs

Task outputs consist of the following BB 3.1 items:

- Pre-Operational Code
- Pre-Operational Test Data
- DPP_3.x
- SWA_2.2
- RAD_1.3, including RNM and RAS
- VVP_1.3
- DDD_1.1
- UTP_1.0
- TRD
- TRRR
- PBR_3.1

## 6.5 Code and Test Data Development Activities

Step 9 activities include:

1) Write pre-operational code
2) Develop unit test data
3) Develop unit test plan
4) Prepare for TRR
5) Conduct TRR

### 6.5.1 Write Pre-Operational Code

**Development Programmers** write the pre-operational code to implement the detailed design, following coding standards provided in TD-11.1 (FORTRAN code) and TD-11.2 (C code). Pre-operational code consists of the system components in the detailed design (software units and sub-units). This code will be written and debugged, and ready for unit testing, but not formally tested.

Use the SWA and DDDs (PDL and file descriptions) as a blueprint for writing the code.

It is recommended that diagnostic code be added to the unit code to facilitate the verification of functionality and outputs. If this is done, two versions of the unit code should be created. The first version will contain the diagnostic code. The second version, with all diagnostic code stripped out, is the baseline pre-operational code.

The code should be debugged until it runs without errors and produces the designed output. Debugging should be accomplished by reviewing compiler error messages and making the indicated corrections. When the code compiles without errors, it should be built into executables in the test environment and run to verify that there are no runtime errors. For this reason, it is recommended that the pre-operational code and test data (c.f. Section 6.5.2) be developed concurrently.

In the process of writing and debugging the pre-operational code, problems with the detailed design may be discovered. In that case, the **Development Scientists** and **Development Programmers** should determine whether the detailed design needs revision to correct the problems. If the revisions are relatively minor and do not affect the Detailed Design Allocation, as is usually the case, document the revisions in updates of the SWA (v2r2) and/or the DDD (v1r1). If the revisions do affect the Detailed Design Allocation, trace the affected system components or product components back to the driving requirements to ensure that these are not compromised by the design revisions.

Update the RAD and its Appendices (RNM and RAS) to v1r3 to capture any changes to the Detailed Design Allocation that have occurred as a result of design revisions during step 9.

When the procedure to build the test configuration into a test executable has been determined (c.f. Section 6.5.3), implement this procedure to verify its correct functionality. A set of unit test executables should be built to verify unit test readiness. At the discretion of the development team, these executables may be run as a unit test “dry run”. A dry run can be useful in identifying problems that require a revision of the unit test plan and/or test items. The process of dry runs plus revision should not be extensive, though. Most of the effort to test and refine the code should occur during step 10. In particular, the TRR should not be delayed by this process.

### 6.5.2 Develop Unit Test Data

**Development Testers** and **Development Scientists** collaborate to produce test data for unit testing of the pre-operational code. The unit test data may be acquired from the input data providers (real data). Real data may be static (acquired in advance of the unit tests and stored in the test environment) or dynamic (acquired in real time during the unit tests). Data may also be acquired from alternative input data sources (proxy data). Proxy data may also be static or dynamic. Data may also be constructed (synthetic data).

Real data is preferable, when available. If real data is not available, proxy data is often an acceptable alternative.

Synthetic data is usually the least preferable, but there may be cases where synthetic data is called for. Usually, this is to construct test cases that will stress the performance of the algorithm or to ensure that all required geophysical conditions are tested.

Dynamic data has the advantage of best simulating real time operations, where the characteristics of the input data is unpredictable, but static data is useful to endure that the test data represents all required geophysical conditions. It is recommended that a combination of real data and static data be used.

The input test data files should have the content and format that are described in the DDDs. It is a good idea to insert intermediate WRITE statements in the code to write out samples of the contents of the input files for verification of the content and format. This is another reason that the pre-operational code and test data be developed concurrently.

Test data should include “truth” data. Truth data sets typically contain the values of environmental or weather products that are traceable to performance requirements (e.g., Atmospheric Vertical Temperature profiles). Truth data sets may be real, proxy or simulated data.

### 6.5.3 Develop Unit Test Plan

**Development Programmers, Development Scientists and Development Testers** collaborate in the development of a unit test plan, documented in UTP v1r0, following guidelines in DG-9.1.

Explain the purpose of each unit, the role of the unit in the product processing system, the major functional steps, and how these steps satisfy the purpose of the unit. This information should be obtained from the SWA and DDD.

Identify all unit components that have been selected for testing. These items should be part of the system architecture as documented in the SWA. Typically, they are sub-processes of the unit's process flow. They should also be identified as verification items in the project's VVP. It is important that the development team confirm that the planned unit test items are documented as verification items in the VVP. Discrepancies must be resolved and either the UTP or the VVP revised as necessary.

Identify the requirements allocated to each test item, consistent with the Detailed Design Allocation.

Determine and describe the environment in which the unit tests will be performed. A project may use the development environment to test the pre-operational system or it may choose to establish a separate test environment. Project constraints will usually determine this choice. For example, operations may request that the test environment be a clone of the operational environment, but cost factors may exclude establishing the development environment as an operational clone. In that case, the best solution may be to use a small operational clone environment as a separate test environment. A project may also choose to perform its pre-operational code unit tests in the development environment and then perform its system integration tests in an operational clone environment. In any case, these choices should be explicitly stated as requirements for the test environment.

Identify all configuration items that will be used in the unit test, including code modules, test data sets, utilities, libraries, etc. The set of configuration items is called the test configuration. Identify the procedure to build the test configuration into a test executable.

Determine and describe the methods that will be used to test each test item. Test methods should be closely related to the verification methods that are documented in the VVP. It is permissible to insert relevant material from the VVP into the UTP, provided it is referenced appropriately. Alternatively, the UTP developers may choose to leave the specific material out of the UTP and refer to specific sections of the VVP that pertain to the test methods for a unit. The latter choice is recommended if the TRR reviewers are already familiar with the material in the project's VVP. It is recommended that the UTP developers consult with the TRR reviewers before deciding how to document test methods in the UTP.

Describe the planned sequence of test actions in sufficient detail that a reviewer can confirm that all test items are exercised, all test data are utilized, all planned test methods are used as planned, and the planned output will allow a reviewer to confirm that the requirements will be satisfied. Note which sequence steps exercise which test items, utilize which test data sets, and use which test methods.

Describe the expected output from each sequence step. The expected output should be described in sufficient detail for unit test reviewers to be able to confirm that the actual unit test output matches the expected output. Output includes runtime messages, diagnostic messages and the content of data files.

State the success criteria for each unit test. Success criteria should include the following:

- All test sequence steps run as planned in the UTP
- Program run time meets requirements
- All runtime message and diagnostic messages are written as expected in the UTP
- The contents of all diagnostic files are as expected in the UTP
- Unit test output satisfies all quantitative performance requirements

For each unit, determine and note which other units must be run to supply input data for the unit test. It is common for the software units to be linked in a sequential order. The SWA and DDD will contain this information.

### 6.5.4 Prepare for TRR

The TRR review lead (**Technical Review Lead)** and review team (**Technical Reviewers)** should have been selected during steps 5, 6, 7, or 8, and listed in the DPP. If this selection was not completed in step 8, the **STAR Branch Chief**, in consultation with the **Development Lead**, should make this selection as soon as possible during step 9.

**Development Scientists** and **Development Testers** assist in a revision of the VVP, following the guidelines in DG-6.3. VVP v1r3 adds to v1r2 by updating the listing and description of verification and validation items and plans, based on changes to the Detailed Design Allocation since CDR, as documented in RAD v1r3.

Cost and schedule variance will often occur during step 9. Therefore, the EPL process calls for an update of the project plan during this step. The **Development Lead** should produce DPP version 3, with assistance from the **Development Scientists, Development Testers,** and **Development Programmers**.

The **Development Lead** leads the preparation of the TRR presentation. The TRR slide package is the Test Readiness Document (TRD). The TRD is prepared by the **Development Lead**, **Development Scientists, Development Testers,** and **Development Programmers,** in accordance with TRD guidelines DG-9.2. DG-9.2.A provides TRD slide templates that can be adapted for the project's TRD. The TRD developers should examine the DPP to determine whether the TRR objectives, entry criteria, exit criteria and/or CLI have been tailored. If so, the TRD slide templates must be adapted to accommodate the tailoring. The TRD developers should use the project's CDD as a source for TRD slides, as many CDD slides can be re-used or adapted.

The **Development Lead,** assisted by the **Development Scientists, Development Testers,** and **Development Programmers,** updates the status of the project risks and associated risk mitigation actions for inclusion in the TRD. Risk management guidelines can be found in PG-1.

The **Development Lead** determines which members of the development team will present the TRD sections. These presenters should be noted in Section Title slides. See DG-9.2.A for examples.

**STAR CM/DM** inserts the standard BB 3.0 items in the project baseline, and updates the Project Baseline Report (PBR) to version 3.0, in accordance with PBR guidelines DG-5.4.

The **Development Lead** informs the **STAR Web Developer** that the TRR artifacts are ready for posting on the STAR EPL website. The **STAR Web Developer** works with **STAR CM/DM** to acquire the project baseline items and post them on the website.

Once these are posted the **STAR Web Developer** informs the **Development Lead**, who then informs the **Technical Review Lead** that the TRR artifacts are available for review. The **Technical Review Lead** then informs all review team members that the artifacts are available to them.

The **Technical Review Lead** and **Technical Reviewers** may at their discretion examine the artifacts and communicate issues to the **Development Lead** prior to the review date, so that the artifacts and/or review presentation may be revised to respond to reviewer concerns.

### 6.5.5 Conduct TRR

The TRR consists of the presentation of the pre-operational code, test data, and test plan by the development team (**Development Lead**, **Development Scientists, Development Testers,** and **Development Programmers)** and the disposition of the review CLI, including entry and exit criteria, by the reviewers (**Technical Review Lead** and **Technical Reviewers**).

The **Technical Review Lead** and the **Technical Reviewers** conduct the TRR to determine whether the pre-operational code conforms to coding standards and is ready for unit testing. Reviewers should be familiar with the TRR guidelines (PRG-9) and check list (CL-9).

The TRR reviewers complete a Test Readiness Review Report (TRRR), following guidelines in DG-9.3. The TRRR will include the reviewers' assessment of the status of the TRR artifacts, the project risks, and associated risk mitigation actions, and an Appendix that consists of the reviewers' disposition of each TRR CLI.

On the basis of its disposition of the TRR CLI, the **Technical Review Lead** and the **Technical Reviewers** determine whether the project is ready to proceed to the next step, "Code Test and Refinement". If not, the TRRR should direct the **Development Lead** to revise the TRR artifacts through specified actions. These actions may include a new assessment of revised TRR artifacts at a delta review.

If a delta TRR is required, the **Development Lead** and support team upgrade the TRR artifacts as requested by the TRR reviewers and present them at a delta TRR. This is repeated until the **Technical Reviewers** pass the project to the next step.

If a delta review is not required, the revision of the TRR artifacts will be deferred to actions performed during step 10 for review at the Code Test Review (CTR), or during later steps for review at later reviews. All of this should be documented in the final version of the TRRR.

**STAR QA** verifies that the TRR was conducted in accordance with STAR EPL standards.

**STAR CM/DM** updates the project baseline via BB 3.1, and updates the Project Baseline Report (PBR) to version 3.1, in accordance with PBR guidelines DG-5.4. BB 3.1 will include all post-TRR revisions to the TRR artifacts, the TRRR, and PBR_3.1.

When the project passes its TRR, the project proceeds to code unit testing and refinement. The final version of the TRRR should include approval for the project to proceed to step 10, and will indicate all open actions that have been deferred to step 10.

**Each stakeholder** who performed activities during step 9 is encouraged to document an assessment of the experience in a personal record. This assessment should include: what was good, what was bad, what worked, what did not work, what can be improved, how it can be improved.

The **Development Lead** should remind the stakeholders to do this. At the conclusion of Development (step 11), the **Development Lead** will collect the final edited personal stakeholder records and incorporate them into a Development Project Report (DPR).

END OF DOCUMENT