NOAA NESDIS
CENTER for SATELLITE APPLICATIONS and RESEARCH

DOCUMENT GUIDELINE

DG-10.1
UNIT TEST REPORT GUIDELINE

Version 3.0

TITLE: DG-10.1: UNIT TEST REPORT GUIDELINE VERSION 3.0

AUTHORS:

Ken Jensen (Raytheon Information Solutions)

UNIT TEST REPORT GUIDELINE
VERSION HISTORY SUMMARY

Version	Description	Revised Sections	Date
1.0	No version 1.0.		
2.0	New Document Guideline (DG-13.3) adapted from CMMI guidelines by Ken Jensen (Raytheon Information Solutions)	New Document	11/19/2007
3.0	Renamed DG-10.1 and revised by Ken Jensen (RIS) for version 3.		10/1/2009

TABLE OF CONTENTS

Page

LIST OF ACRONYMS

CDR	Critical Design Review
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
CMMI	Capability Maturity Model Integration
CREST	Cooperative Remote Sensing and Technology Center
CTR	Code Test Review
DG	Document Guideline
EPL	Enterprise Product Lifecycle
NESDIS	National Environmental Satellite, Data, and Information Service
NOAA	National Oceanic and Atmospheric Administration
PAR	Process Asset Repository
PDR	Preliminary Design Review
PG	Process Guideline
PRG	Peer Review Guideline
PRR	Project Requirements Review
QA	Quality Assurance
RAD	Requirements Allocation Document
SG	Stakeholder Guideline
STAR	Center for Satellite Applications and Research
SWA	Software Architecture Document
TD	Training Document
TG	Task Guideline
UTP	Unit Test Plan
UTR	Unit Test Report
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, CIMSS, CIOSS, CIRA, CREST) distributed throughout the US, multiple support contractors, and NESDIS Operations.

NESDIS/STAR is implementing an increased level of process maturity to support the exchange of these software systems from one location or platform to another. The Unit Test Report (UTR) is one component of this process.

1.1. Objective

The objective of this Document Guideline (DG) is to provide STAR standards for the UTR. The intended users of this DG are the personnel assigned by the Development Lead to the task of creating a UTR for the project.

1.2. The Unit Test Report

The UTR is a critical artifact for a project's Code Test Review (CTR)[1]. It documents the results of testing of each software unit to verify that the requirements allocated to the unit's software components are satisfied.

A separate UTR is produced for each distinct project in the STAR Enterprise.

The intended target audiences are customers, product users, and unit test reviewers. Typically, the UTR is prepared by the project's development team, under the direction of the Development Lead and in consultation with quality assurance (QA) and test personnel.

The UTR should be developed as a Microsoft Word document. Upon approval, the approved version of the UTR may be converted to an Adobe pdf file for storage in the project artifact repository.

[1] Refer to the STAR EPL Process Guidelines (PG-1 and PG-1.A) for a description of the STAR EPL gates and reviews.

1.3. Background

This DG defines guidelines for producing a UTR. This DG has been adapted from Capability Maturity Model Integration (CMMI) guidelines (CMMI-DEV-v1.2, 2006). It has been tailored to fit the STAR Enterprise Product Lifecycle (EPL) [2] process.

1.4. Benefits

A UTR developed in accordance with the standards in this DG assists the development team to demonstrate to reviewers, users, and operations that the product processing system is ready to be installed in the operations environment. It is therefore a requirement that a UTR be developed in accordance with the guidelines in this document. The UTR will be reviewed at the CTR to determine whether the project is ready for product integration and system testing.

1.5. Overview

This DG contains the following sections:

Section 1.0 -	Introduction
Section 2.0 -	References
Section 3.0 -	Standard Table of Contents
Section 4.0 -	Section Guidelines
Appendix A -	Examples
Appendix B -	Templates

[2] For a description of the STAR EPL, refer to the STAR EPL Process Guidelines (PG-1 and PG-1.A).

2. REFERENCE DOCUMENTS

RAD: Requirements Allocation Document is a project artifact that is produced by the Project Team for the Project Requirements Review (PRR). Revisions are customarily made for Preliminary Design Review (PDR) and Critical Design Review (CDR). The UTR developer needs this to prepare the Requirements Allocation sections of the UTR. This document will be available to approved users in the project artifact repository.

VVP: Verification and Validation Plan is a project artifact that 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. The UTR developer will find this useful in preparing the Unit Test Description sections of the UTR. This document will be available to approved users in the project artifact repository.

UTP: Unit Test Plan, a project artifact produced by the product development team, is the most important reference document for the UTR developer. The UTR developer needs this to prepare the Software Units and Unit Test Results sections of the UTR. This document will be available to approved users in the project artifact repository.

SWA: Software Architecture Document is a project artifact produced by the product development team. The UTR developer needs this to prepare the Software Units and Unit Test Results sections of the UTR. This document will be available to approved users in the project artifact repository.

All of the following references are STAR EPL process assets that are accessible in a STAR EPL Process Asset Repository (PAR) on the STAR web site:

http://www.star.nesdis.noaa.gov/star/EPL_index.php.

PG-1: STAR EPL Process Guideline provides the definitive description of the standard set of processes of the STAR EPL.

PG-1.A: STAR EPL Process Guideline Appendix, an appendix to PG-1, is a Microsoft Excel file that contains the STAR EPL process matrix (Stakeholder/Process Step matrix), listings of the process assets and standard artifacts, descriptions of process gates and reviews, and descriptions of stakeholder roles and functions.

PRG-10: Code Test Review Guidelines are the guidelines for the CTR. It is useful for the developer of the UTR to understand what the reviewers will expect when reviewing the UTR.

CL-10: Code Test Review Check List is the check list for the CTR. It is useful for the developer of the UTR to understand the specific Check List items (CLI) that the reviewers of the UTR will be required to approve.

DG-0.1: STAR Document Style Guideline is a STAR EPL Document Guideline (DG) that provides STAR standards for the style and appearance of STAR documents developed as Microsoft Word files

SG-14: STAR EPL Development Scientist Guidelines provides a description of standard tasks for Development Scientists, including development of the UTR.

SG-15: STAR EPL Development Tester Guidelines provides a description of standard tasks for Development Testers, including development of the UTR.

SG-16: STAR EPL Development Programmer Guidelines provides a description of standard tasks for Development Programmers, including development of the UTR.

TG-10: STAR EPL Code Test and Refinement Task Guidelines provides a description of standard tasks for process step 10, during which the UTR is developed.

3. STANDARD TABLE OF CONTENTS

4.3.4 Test Sequence

4.3.5 Success Criteria

4.4 Test Results

4.5 Limitations and Risks

…………………………………….

N+2.0 <UNIT N NAME> UNIT TEST RESULTS

N+3.0 LIST OF REFERENCES

4. SECTION GUIDELINES

This section contains the STAR guidelines for each section of the UTR.

The UTR should follow the STAR standard for style and appearance, as stated in DG-0.1.

4.1. Table of Contents

The Table of Contents can be inserted by using Word's Insert → Reference → Index and Tables → Table of Contents function or by pasting the Table of Contents from this DG into your document and updating it for the section headers you make for your document. Use a page break if necessary to ensure that the Table of Contents appears at the top of a page.

4.2. List of Figures

A List of Figures should be provided after the Table of Contents. A page break should be used if necessary to ensure that the List of Figures appears at the top of a page. To create a List of Figures, use Word's Insert → Reference → Index and Tables → Table of Figures function, selecting the "Table of Figures" Style. Alternatively, the List of Figures can be created by pasting the List of Figures for this DG into your document.

Figures should be created by using Word's Insert → Picture → From File function or Word's Insert → Object function. Figures should be numbered X.Y, where X is the main section number where the figure resides and Y = 1,N is the ordered number of the figure in the section. Figure captions should have Arial bold 12 point font, should be center justified, and should have a "Table of Figures" Style. A Figure Caption template is provided in Appendix B of this DG.

4.3. List of Tables

A List of Tables should be provided after the List of Figures. The List of Tables can appear on the same page as the List of Figures, with three blank lines separating them, provided both lists can fit on the same page. If both lists cannot fit on the same page, a page break should be used to ensure that the List of Tables appears at the top of a page.

To create a List of Tables, use Word's Insert → Reference → Index and Tables → Table of Figures function, selecting the "Table - Header" Style. Alternatively, the List of Tables can be created by pasting the List of Tables for this DG into your document.

Tables should be created with the Table → Insert → Table function. Tables should be numbered X.Y, where X is the main section number where the table resides and Y = 1,N is the ordered number of the table in the section. Table titles should have Arial bold 12 point font, should be center justified, and should have a “Table - Header” Style. A Table Title template is provided in Appendix B of this DG. Table text should have Arial regular 10 point font.

4.4. List of Acronyms

The use of acronyms is encouraged. A two word or longer name for an item (e.g., Unit Test Report) should be given an acronym (e.g., UTR) if the name is used more than once in the document. A List of Acronyms should be provided after the List of Tables. The List of Acronyms should be in alphanumeric order. Use the List of Acronyms in this DG as a template. A page break should be used if necessary to ensure that the List of Acronyms appears at the top of a page.

4.5. Section 1 – Introduction

The UTR shall include an Introduction Section. This section shall include

- A well-defined purpose and function for the document
- Specific intended user(s)
- How the intended user(s) should use the document
- A responsible entity for generating the document
- A responsible entity for review/approval of the document
- A responsible entity for storage, accessibility, and dissemination
- A brief overview of the contents of each main section

4.6. Section 2 – Software Units

This is a brief section whose purpose is to list the product processing system’s software units whose test results will be described in the sections to follow. The software units are the primary components of the product processing software architecture subset of the system architecture. They are documented in the project’s Software Architecture Document

(SWA). This section should only contain the list of software units and a System-Layer process flow diagram, also known as a Layer-1 data flow diagram. Each software unit that will be tested should be highlighted in the diagram. The list of software units and the Layer-1 diagram should be obtained from Section 2 of the latest version of the UTP.

4.7. Section 3 – <Unit 1 Name> Unit Test Results

Each unit's test results are documented in its own section. Section 3 is reserved for the first unit test, Section 4 for the second unit test, etc. It is recommended that the order of the units in the UTP matches the sequential order of the processing system. This order should be reflected in numerical identifications assigned to the units in the system architecture, as documented in the SWA and UTP, and as listed in Section 2 of the UTR.

The guidelines for each unit's section (Section 3, 4, etc., of the UTR) are identical. Each section should contain the following subsections:

1) Requirements Allocation
2) Process Flow
3) Unit Test Description
4) Test Results
5) Limitations and Risks

The following guidelines are provided for subsections 3.1 - 3.5. It is understood that these apply to every unit's section (Section 4, 5, etc.).

4.7.1. Section 3.1 – Requirements Allocation

Identify the requirements that have been allocated to components of the unit's software architecture. This information should be obtained from the RAD and the UTP. Requirements should be numbered as in the RAD and UTP.

For each identified requirement, list the components of the unit's software architecture that are traceable to the requirement, as documented in the RAD and UTP.

4.7.2. Section 3.2 – Process Flow

Show the unit process flow as a process flow diagram, also known as a Layer-2 data flow diagram. This diagram should be obtained from Section 3.2 of the UTP.

4.7.3. Section 3.3 – Unit Test Description

Describe the tests that have been performed on the software unit. Information in this section should draw heavily from, and be consistent with, the project UTP. Subsections should include Purpose and Function, Test Items, Test Data Description, Test Sequence, Success Criteria, and Risks.

- The subsection for Purpose and Function should explain the purpose of the 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 explanation should be obtained from the UTP. The content should be primarily textual. References to appropriate figures and tables in the SWA and UTP can be made.

- The subsection for Test Items should identify all unit components that have been selected for testing. These items should be traceable to the requirements allocation items presented in Section 3.1 of the UTR. Typically, they are sub-processes of the unit's process flow that is shown in Section 3.2. They should also be identified as verification items in the project's VVP. Test items should be numbered as they are numbered in the UTP.

- The subsection for Test Data Description should list and describe all data files that will be used as input files for the unit test. Obtain this information from the UTP. If the UTP presents these items in one or more tables, copy the tables into this subsection of the UTR. Files to be listed here include:
    - "Test data". These data sets include the sensor data (real, proxy, or simulated), ancillary data, control files, parameter files, and look up tables that are needed to run the unit test.
    - "Truth" data. These are data sets that will be used to assess the quality of the unit's output. Truth data sets typically contain the values of environmental or weather products that are traceable to performance requirements.

- The subsection for Test Sequence should describe the planned sequence of test actions in sufficient detail that a reviewer can confirm that all test items are exercised and all test data are utilized. Obtain this information from the UTP. If the UTP presents the test sequence in one or more tables, copy the tables into this subsection of the UTR.

- The subsection for Success Criteria should state the success criteria for the unit test. Obtain this information from the UTP.

- The subsection for Risks should identify and evaluate risks to successful implementation of the test plan for this unit. Obtain this information from the UTP.

4.7.4. Section 3.4 – Test Results

Describe the results of the tests that were described in Section 3.3 in a way that demonstrates the verification of the requirements allocated to components of the software unit. Show how the results demonstrate that the requirements allocated to the software units are satisfied. Note any requirements allocations whose verification is incomplete or questionable. Note that these will be discussed in Section 3.5 of the UTR.

The expected unit test results should be documented in the UTP. Ideally, this section of the UTR will compare the actual test results with the expected results, noting any gaps or deficiencies. The expected output includes runtime messages, diagnostic messages and the content of output data files.

Runtime messages are messages written by the operating system to a runtime log file or other designated output source (e.g., a monitor connected to the computer from which the program execution command has been entered). These may occur if the unit code is written to generate such messages as a way to test functionality. Describe the exact content of each expected runtime message and at what point in the test sequence it is expected.

Diagnostic messages are messages written by the unit program to a runtime log file or other designated output source (e.g., a monitor connected to the computer from which the program execution command has been entered). The nominal purpose of a diagnostic message is to report a functional result (e.g., ‘subroutine X called’) or the quantitative value of an input, intermediate, or output variable (e.g., ‘X(50) = 7’). Describe the exact content of each expected diagnostic message and at what point in the test sequence it is expected.

Data files include the output data sets that are designed to be produced by the unit program. In addition, a diagnostic program may write intermediate data sets to diagnostic files. Describe the expected content of all diagnostic files that are sampled as part of the unit test. Use figures, graphs, tables, scatter plots as warranted to display the comparison of actual results to expected results.

Note expected run times. Demonstrate that run times meet operational run time requirements. This demonstration can include scaling factors if the test environment is predicted to be slower than the operational environment by a known amount.

4.7.5. Section 3.5 – Limitations and Risks

Discuss any of the unit's requirements allocations whose verification is incomplete or questionable, due to gaps or deficiencies between the actual test results and the expected test results that were noted in Section 3.4 of the UTR.

Identify risks that are generated by these limitations. For each identified risk:

- Recommend a plan to mitigate the risk
- Note risk closure criteria
- Recommend actions to implement the risk mitigation plan.
- Note criteria to close actions

4.8. List of References

This section should consist of a List of References that includes all references cited in the document. Include all references deemed useful by the Product Team. References should be listed in alphabetical order. References that begin with an author list should begin with the last name of the lead author. A template is provided in Appendix B.

APPENDIX A. EXAMPLES

An example of a UTR that follows the STAR standards and guidelines will be developed and placed in the STAR EPL PAR.

APPENDIX B. TEMPLATES

This appendix contains templates for specific pages and sections of the UTR.

B.1 Cover Page Template:

In this template, <X> = 1.0 for version 1, <X> = 1.1 for version 1 revision 1, <X> = 2.0 for version 2 etc. <Project Name> should be the actual approved name of the Project.

NOAA NESDIS
CENTER for SATELLITE APPLICATIONS and RESEARCH

<PROJECT NAME>
UNIT TEST REPORT

Version <X>

B.2 Document Header Template:

In this template, <X> = 1.0 for version 1, <X> = 1.1 for version 1 revision 1, <X> = 2.0 for version 2 etc.

In this template, <Project Name> should be the actual approved name of the Project.

In this template, <Y> = the actual page number.

In this template, <Z> = the actual total number of pages

NOAA/NESDIS/STAR

UNIT TEST REPORT
Version: <X>
Date: <Date of Latest Signature Approval>

<Project Name>
Unit Test Report

Page <Y> of <Z>

B.3 Document Cover Page Footer Template:

Hardcopy Uncontrolled

B.4 Document Footer Template:

Hardcopy Uncontrolled

B.5 Approval Page Template:

In this template, <X> = 1.0 for version 1, <X> = 1.1 for version 1 revision 1, <X> = 2.0 for version 2 etc. <Project Name> should be the actual approved name of the Project.

TITLE: <PROJECT NAME> UNIT TEST REPORT VERSION <X>

AUTHORS:

<Lead Author>

<Co-Author 1>

<Co-Author 2>

<etc.>

APPROVAL SIGNATURES:

____ <Actual Signature Date>
<Name of Project Development Lead> Date
Project Development Lead

____ <Actual Signature Date>
<Name of Project Manager> Date
Project Manager

____ <Actual Signature Date>
<Name of Agency Approver> Date
Agency

B.6 Version History Page Template:

In this template, <Project Name> should be the actual approved name of the Project.

<PROJECT NAME>
UNIT TEST REPORT
VERSION HISTORY SUMMARY

Version	Description	Revised Sections	Date
1.0	Created by <Name of Developer(s)> of <Name of Developers' Agency/Company> for Code Test Review (CTR).	New Document	<Actual date of Latest approval signature>
1.1	[As needed] Revised by <Name of Developer(s)> of <Name of Developers' Agency/Company> to describe changes due to new unit tests and/or in response to CTR actions	<applicable sections>	<Actual date of Latest approval signature>
1.2	Ditto	Ditto	Ditto
etc.			

B.7 Figure Caption Template:

Figure X.Y - <Figure caption in Arial regular 12 point font>

B.8 Table Title Template:

Table 4.5 - <Table title in Arial regular 12 point font>

B.9 List of References Template:

Ackerman, S. *et al.* (1997). Discriminating clear-sky from cloud with MODIS: Algorithm Theoretical Basis Document, Version 3.2.

Asrar, G., M. Fuchs, E. T. Kanemasu, and J. L. Hatfield (1984). Estimating absorbed photosynthetically active radiation and leaf area index from spectral reflectance in wheat. *Agron. J.*, 76:300-306.

Bauer, E., and Kohavi, R., (1998). An empirical comparison of voting classification algorithms: bagging, boosting, and variants, *Machine Learning*, **5**: 1-38.

Bonan, G.B. (1995). Land-atmosphere interactions for climate system models: Coupling biophysical, biogeochemical, and ecosystem dynamical processes. *Remote Sens. Environ.*, 51:57-73.

Food and Agriculture Organization of the United Nations*, Digital Soil Map of the World and Derived Soil Properties-Version 3.5*, FAO/UNESCO, Rome, 1995.

Friedl, M. A., and C.E. Brodley (1997). Decision tree classification of land cover from remotely sensed data. *Remote Sens. Environ.*, 61:399-409.

Scepan, J. (1999), Thematic validation of high-resolution global land-cover data sets. *Photogramm. Eng. Remote Sens.*, 65:1051-1060.

Shukla, J., C. Nobre, and P. Sellers (1990). Amazon deforestation and climate change. *Science*, 247:1322-1325.

Wilson, M.F., and A. Henderson-Sellers (1985). A global archive of land cover and soils data for use in general circulation models. *J. Clim*., 5:119-143.

Wu, A., Z. Li, and J. Cihlar (1995). Effects of land cover type and greenness on advanced very high resolution radiometer bidirectional reflectances: analysis and removal. *J. Geophys. Res.*, 100: 9179-9192.

END OF DOCUMENT