NESC Academy Training
Participant Manual

Lesson 1: Introduction

Course Goal

The goal of this course is to present detailed case studies from the career experiences of software expert Michael Aguilar and problem-solving activities involving software as an engineering discipline to a new generation of NASA engineers and scientists, designated contractors, university professors, and graduate students in engineering programs.

Objectives

At the conclusion of this lesson, participants will be able to:

- Understand the objectives of the course.
- Identify the schedule and course requirements.
- Identify the faculty and other participants.
- Use the layout and symbols to navigate through the Participant Manual.
- Provide appropriate feedback using the participant polling devices.

Class Schedule

	TUESDAY MARCH 13	WEDNESDAY MARCH 14	THURSDAY MARCH 15
Morning 8 a.m.– 12 p.m.	Lesson 1—Introduction Lesson 2—Software Introduction Lesson 3—Engineering Examples	Lesson 7—CASE Examples and Exercises Lesson 8—Software Quality Assurance	Lesson 11—Open Panel Session Q&A
Lunch 12–1 p.m.			
Afternoon 1 p.m.– 4 p.m.	Lesson 4—Software Development History Lesson 5—UML Introduction Lesson 6—Software Engineering Tools and Techniques	Lesson 9—Unique NASA Software Quality Issues Lesson 10—NASA Best Practices and Lessons Learned	Lesson 11—Open Panel Session Q&A Lesson 12—Wrap-Up
4:15 p.m.	Reception		

Course Goal and Requirements

The goal of this course is to present detailed case studies from the career experiences of Michael Aguilar and problem-solving activities involving software as an engineering discipline to a new generation of NASA engineers and scientists, designated contractors, university professors, and graduate students in engineering programs.

This course will present elements of the following topics:

- Software introduction.
- Engineering examples.
- Software development history.
- UML introduction.
- Software engineering tools and techniques.
- CASE examples and exercises.
- Software quality assurance.
- Unique NASA software quality issues.
- NASA best practices and lessons learned.
- Open panel session Q&A.

Teaching Methodology

The 20 hours of class time break down as follows:

Lecture and Discussion	15 hours 30 minutes
Activities	3 hours
Participant Evaluation and Discussion	1 hour
Course Evaluation	30 minutes

Michael Aguilar, the lead instructor for this course, intends to maintain a very high level of technical discourse that focuses on the lessons learned from a series of case studies.

Mr. Aguilar will provide you with an opportunity to ask questions and get feedback. You are expected to participate in all classroom activities and problem-solving exercises.

Course Faculty

The course faculty is led by an NESC Discipline Expert (NDE) currently working with the NASA Engineering and Safety Center. Below is a short biographical sketch of the expert who will be involved in teaching this course.

Michael L. Aguilar

Michael Aguilar works at the NASA Goddard Space Flight Center (GSFC) as the NASA Engineering and Safety Center (NESC) Discipline Expert in Software Engineering.

Mr. Aguilar has developed embedded systems his entire career, acting as project manager, configuration manager, real-time performance engineer, hardware API developer, test driver developer, hardware installation and integration engineer, and software installation and integration engineer. In addition to spacecraft command and control systems, he has worked on software development on flight simulators, submersible robotics, nuclear reactor monitoring and control systems, safety-critical embedded software, an NSSN submarine sonar system, a robotic manipulator for hip replacement surgery, RADAR Command and Control Centers, and NATO communications systems.

As the James Webb Space Telescope (JWST) Science Instrument Flight Software Manager, Mr. Aguilar managed the interface and integration of the JWST command and data handling (C&DH) core flight software development at GSFC and the Science Instrument flight software applications developed externally by the European Space Agency, the Canadian Space Agency and EMS Technologies, the University of Arizona and Lockheed Martin ATC, and the JPL/European Consortium.

As the safety engineer for the Interim Control Module (ICM), a robust attitude control system for the International Space Station (ISS), Mr. Aguilar performed the analysis and assessment, both in hardware and software, of the ICM quad-processor autonomous Fault, Detection, Isolation, and Recovery (FDIR) capabilities.

Mr. Aguilar received his master of science degree in software engineering from Carnegie Mellon University in Pittsburgh and his bachelor of science degree in computer science from California State University, Northridge.

Guest Speakers

The following speakers will be speaking on the last day of the course and will participate in the panel discussion.

Kenneth A. Costello

Mr. Costello has been with NASA since 1997. He is currently chief engineer at the Independent Verification and Validation (IV&V) Facility, where he oversees the development and continuous improvement of best practices for technical execution and project management approaches. Mr. Costello has served as the IV&V project manager on several critical NASA projects, including the Deep Impact mission and the Mars Exploration Rovers mission. His main field of interest is implementing modeling for the validation of real-time software systems.

Mr. Costello received a master of science degree in software engineering from West Virginia University. He has a bachelor of science degree in aerospace engineering from Pennsylvania State University.

Dr. Rajeev Joshi

Dr. Joshi is a senior software engineer assigned to the Laboratory for Reliable Software (LaRS) at the Jet Propulsion Laboratory (JPL) in Pasadena, CA. Dr. Joshi's main field of interest is in improving software reliability through the use of automated verification tools. He has built several tools and published research papers on various techniques based on formal methods, including static code analysis, logic model checking, theorem proving, randomized testing, and automatic code generation. He has also published reports on automatic resource configuration, concurrency theory, and information flow.

Dr. Joshi worked at the Compaq/HP Systems Research Center in Palo Alto, CA. before joining JPL in 2003. He worked for 2 years at AT&T's Bell Laboratories in Murray Hill, NJ, prior to graduate school. Dr. Joshi has a doctorate degree in computer science from the University of Texas at Austin.

Dr. John C. Kelly

Dr. Kelly is the Program Executive for Software Engineering within the NASA Headquarters Office of Chief Engineer. He is responsible for the establishment of agency-wide engineering and management policy, guidance, processes, and supporting infrastructure to effectively meet the scientific and technical objectives of software products developed under NASA funding. Dr. Kelly's accomplishments include leadership responsibilities in the development and release of NPR 7150.2, NASA Software

Engineering Requirements; NPD 2820.1, NASA Software Policy; the annual NASA Software Inventory; and the agency's Software Engineering Curriculum Plan.

Previously, Dr. Kelly served as a principal engineer at NASA's Jet Propulsion Laboratory (JPL) and led that agency's initiative in formal methods for computer systems, and software formal inspections.

Dr. Kelly has served as a professor of computer science at Furman University in Greenville, SC, and a mathematics professor at Darton College in Albany, GA. A native of Miami, Dr. Kelly received a bachelor of science degree in computer science, a master of science degree, and doctorate degree all from Florida State University.

Kevin P. Murphy

During his 25 years with Boeing Co., Mr. Murphy has worked on the Space Shuttle and International Space Station (ISS).

Mr. Murphy currently works at Boeing as the Software Architect of the ISS. He also provides systems engineering support for NASA's Constellation project.

In 2002, Boeing selected him to be one of the company's "technical fellows," an honorary title. In that capacity, Mr. Murphy provides consultation services across the company, including missile defense and future combat systems.

Mr. Murphy holds a bachelor of science degree in aerospace engineering from the University of Texas.

Thomas T. Pressburger

Mr. Pressburger is currently assigned to the Robust Software Engineering Group at NASA's Ames Research Center in Moffett Field, CA. The engineering group is part of the Intelligent Systems Division (Code TI).

Since he started at Ames in 1992, Mr. Pressburger has supported projects in program synthesis and transformation systems, which translate high-level specifications to code in such areas as solar system geometry problems (Amphion), Kalman filters (AutoFilter), Bayesian statistical analysis (AutoBayes), and Java Pathfinder 1 (translating Java to the language of the SPIN model checker). He has been involved in process improvement at Ames, participating in the CMMI preappraisals of Ames projects. Mr. Pressburger has also been involved with formulating Constellation requirements and currently participates in contractor oversight for

CEV in the tools and process area as well as simulation and modeling.

For the past year, Mr. Pressburger has been in charge of an OSMA SARP initiative to produce a practitioner's guide to model checking. Mr. Pressburger is an Ames representative to the intercenter Software Working Group.

He earned a master of science degree in computer science from Stanford University and holds a bachelor of science degree in mathematics from the California Institute of Technology.

David M. Pruett

Mr. Pruett spent much of his NASA career working on real-time software projects, including the International Space Station (ISS).

A Texas native who first learned to program in FORTRAN in high school, Mr. Pruett worked at Johnson Space Center (JSC) while attending Texas A&M University in Kingsville, TX (formerly Texas A&T). NASA hired him upon graduation. After a stint in the Navy, Mr. Pruett returned to NASA and JSC in 1973. He was assigned to the ISS software program in 1987.

Mr. Pruett had worked his way to ISS software manager when he retired from NASA in 2001. Rather than walk away from the project, he joined Geocontrol Systems, Inc., a Houston company that was awarded the ISS software contract. Mr. Pruett currently works for Geocontrol Systems out of his home near Lexington, KY.

Mr. Pruett received a master of science degree in software engineering from the University of Houston, Clear Lake, and has a bachelor of science in mathematics.

Garth J. Watney

Garth Watney has been with NASA at JPL for the past 12 years. A member of the Flight Software Applications group, Mr. Watney has supported a number of flight projects, including Cassini, Deep Impact, and the Space Interferometry Mission. Among his special interests are model-based engineering methods, the autogeneration of flight code, runtime verification, and the autogeneration of formal verification models from UML statecharts.

Mr. Watney has a bachelor of science degree in mathematics from UCLA and is currently pursuing a master of science degree in computer science from the University of Southern California.

How to Use This Manual

After you complete the course, you can use this manual to help recall topics covered in the course. During course delivery, you are not expected to follow along using the narrative in this manual. You will, however, be expected to review the directions for the activities it describes.

The manual has three main components:

- *Lesson 1* contains the course schedule, evaluation scheme, mission, and topics. It also explains the layout and organization of your Participant Manual.
- *Lessons 2–12* are content lessons.
- *Slide note* pages contain copies of the PowerPoint slides used in the course and space for taking notes.

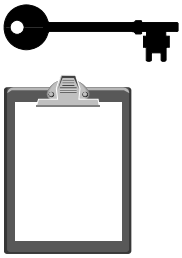
Lesson Titles

The lessons that Mr. Aguilar and his colleagues will discuss are:

- Lesson 2: Software Introduction.
- Lesson 3: Engineering Examples.
- Lesson 4: Software Development History.
- Lesson 5: UML Introduction.
- Lesson 6: Software Engineering Tools and Techniques.
- Lesson 7: CASE Examples and Exercises.
- Lesson 8: Software Quality Assurance.
- Lesson 9: Unique NASA Software Quality Issues.
- Lesson 10: NASA Best Practices and Lessons Learned.
- Lesson 11: Open Panel Session Q&A.
- Lesson 12: Wrap-Up

Symbols

Throughout this manual, two symbols appear in the left margin of the pages.



The key shown at left indicates a key concept in the text.

The clipboard shown at left precedes a list of important items presented in the text.

Course Prerequisites

This course has been designed for NASA scientists and engineers with 2–5 years of experience. Participants are also expected to have completed the precourse comprehension check with a minimum score of 80 percent.

Evaluations

The precourse has two types of evaluation: course and participant. The course evaluation has three parts: precourse, classroom conclusion, and postcourse. The participant evaluation also has three parts: precourse comprehension check, classroom informal assessments, and final check.

Course Evaluation

The precourse evaluation uses Zoomerang, a Web-based tool that presents a series of questions. It takes only a few minutes to complete and provides valuable information to the course designers.

The classroom conclusion evaluation consists of your daily comments on each lesson.

Several weeks after you complete this course, you will receive a postcourse evaluation e-mail asking for your reflections on the course. Like the course evaluation, the postcourse evaluation will use the Zoomerang Web-based tool. It will take only a few minutes to complete. Your feedback is appreciated.

Participant Evaluation

The comprehension check built into the precourse is a short, automatically scored multiple-choice quiz. When you earn a passing score on the quiz, you can print a precourse completion certificate. You can take the quiz as many times as you choose.

During the course, there will be a number of classroom informal assessments to provide you with feedback on your participation in activities and discussion. Your responses to questions will be evaluated by Mr. Aguilar and his colleagues.

The final check at the end of the course will be a multiple-choice test. Your responses will be gathered with participant polling devices. Your course certificate of completion will be awarded on the basis of those results.

Photographs and Other Media

All photographs, drawings, illustrations, schematics, animations, and videos used in this course are courtesy of NASA, unless otherwise noted.

Summary

This lesson has identified the objectives of this course, acquainted you with the course requirements and schedule, introduced you to the faculty, explained how to use this manual, and described the course feedback mechanisms. The next lesson will give an introduction to software.

Individual/Small Group
10 minutes

Activity 1-1

Polling the Participants

Purpose

This activity is designed to give you an opportunity to tell the faculty and other participants about your background and interests. It will also introduce you to using the participant polling device (PPD) as a method of increasing interaction within the class. In addition, it will offer you an early interaction with the course content.

Directions

1. You will use your participant polling device (PPD) that you were issued at the beginning of class in this activity.
2. When you use your PPD, make sure to position yourself so that you have a clear visual path to the receiving unit.
3. You will be asked two sets of questions: background questions about yourself and questions to stimulate your thinking about the engineering challenges of the software discipline.
4. When you answer a question on the screen, note that your PPD number will change color on the screen to indicate that your answer has been recorded.
5. Answer the remaining questions as the instructor presents them.