Conf-931160--38

WHC-SA-1977-FP

# An Approach to Software Quality Assurance for Robotic Inspection Systems

G. R. Kiebel

Date Published
October 1993

To be presented at
American Nuclear Society
1993 Winter Meeting
San Francisco, California
November 14-19, 1993

Prepared for the U.S. Department of Energy
Office of Environmental Restoration and
Waste Management

Westinghouse
Hanford Company Richland, Washington

Hanford Operations and Engineering Contractor for the
U.S. Department of Energy under Contract DE-AC06-87RL10930

Approved for Public Release

## LEGAL DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or any third party's use or the results of such use of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof or its contractors or subcontractors. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

This report has been reproduced from the best available copy.

Printed in the United States of America

# AN APPROACH TO SOFTWARE QUALITY ASSURANCE FOR ROBOTIC INSPECTION SYSTEMS

Gary R. Kiebel
Westinghouse Hanford Company
Box 1970, M.S. N1-42
Richland, WA, 99352
(509) 376-4995

## ABSTRACT

Software quality assurance (SQA) for robotic systems used in nuclear waste applications is vital to ensure that the systems operate safely and reliably and pose a minimum risk to humans and the environment. This paper describes the SQA approach for the control and data acquisition system for a robotic system being developed for remote surveillance and inspection of underground storage tanks (UST) at the Hanford Site.

## INTRODUCTION

The Light-Duty Utility Arm (LDUA) System has a multi-axis robotic manipulator arm with a 2.74-m (9-ft) reach and 22.7-kg (50-lb) payload that is mounted on the end of a 13.7-m (45-ft) positioning mast. It is designed to enter a UST through an available 30.5-cm (12-in.) riser.

A deployment vehicle carries the positioning equipment to insert the mast and manipulator arm into the riser, and carries a containment enclosure to control contamination when the manipulator arm is in the tank or being transported from tank to tank. The system is controlled remotely from an operations and control trailer located outside the perimeter fence of the tank farm, which is up to 275 m (900 ft) from the deployment vehicle.

A set of interchangeable end effectors is carried on the end of the manipulator arm. These end effectors provide a wide range of observation and measurement functions such as photographic and video inspection and recording, detailed surface mapping of the tank and surface of the waste, and insitu chemical analysis of the waste.

The mast/manipulator arm and each end effector have a subsystem controller responsible for controlling their basic function and providing some amount of stand-alone operation. There is also a supervisory control and data acquisition system that operates at a level above all the subsystem controllers. This supervisory level integrates operation of the mast/manipulator arm and end effectors and provides centralized data management.

The design of the LDUA control and data acquisition system is based on experience and technology gained from the Robotic Technology Development Program (RTDP) of the Office of Technology Development (OTD) of the U.S. Department of Energy (DOE).

The LDUA is being developed under the sponsorship of the UST-Integrated Demonstration (UST-ID) Program of the OTD. It is the goal of the UST-ID program to demonstrate emerging technologies that are useful to characterization and remediation of USTs. Much of the work to date has been oriented toward research and the production of prototypic systems for cold (nonradioactive) demonstrations. The LDUA system will be one of the first fully qualified for hot deployment.

The component parts of the LDUA are being supplied by a consortium of DOE sites and national laboratories, some of whom are in partnership with commercial suppliers. Westinghouse Hanford Company (WHC), the operations and engineering contractor at the Hanford Site, is providing the overall project coordination and system integration. The robot and its low-level control system are being supplied by a commercial vendor. Sandia National Laboratory is supplying the supervisory control and data acquisition system. End effectors are being supplied by Oak Ridge National Laboratory, Pacific Northwest Laboratory, Savanna River Technology Center, and Idaho National Engineering Laboratory.

## THE CHALLENGE

Several fundamental issues in choosing an SQA approach for the LDUA were apparent from the beginning. LDUA software would have to meet WHC quality assurance requirements because the application demanded it. Documented evidence of quality would be provided to the organizations responsible for operations at the Hanford Site tank farms for approval of the system for use in the USTs. The use of existing RTDP software with no SQA documentation would have to be qualified. Many software suppliers had no experience or expertise with SQA, but would have to produce software to meet WHC SQA standards. Finally, as with any project, resources were finite.

## THE PHILOSOPHY

Prior experience with software projects at WHC provided some guiding principles used in establishing the LDUA SQA approach. The most important of these principals were: (1) major benefits come from a small set of well-chosen SQA provisions that are consistently applied; (2) an overly ambitious plan will collapse under its own weight; and (3) buy-in from the implementors is essential to the success of any SQA program.

There is no lack of templates for SQA plans, but there is no "one-size-fits-all" template. Effective plans must be tailored to the specific project and as carefully designed as any part of the system. In order to progress from a generalized template to a more focused plan, the following possible quality goals for the LDUA software were examined:

- comprehensiveness
- efficiency
- expendability
- flexibility
- integrity
- interoperability

- maintainability
- portability
- reliability
- responsiveness
- reusability
- survivability
- testability
- usability.

The primary SQA goals of the software to be used with the LDUA were reliability, usability, and maintainability. Reliability was important because significant operating costs and potential radiation exposure to workers could be the consequence of frequent system breakdowns. Maintainability was important because WHC would ultimately be responsible for repairing and improving the software of which much was custom that would not be supported by its supplier after delivery to WHC. Usability was important because the LDUA is to be operated by operations personnel rather than engineers and scientists.

## THE BASIS

Westinghouse Hanford Company SQA requirements derive from the American Society of Mechanical Engineers (ASME) NQA-1, "Quality Assurance Program Requirements for Nuclear Facilities"[1] and ASME NQA-2a, Part 2.7 "Quality Assurance Requirements of Computer Software for Nuclear Facility Applications."[2]

At a minimum, quality-affecting software at WHC must have the following:

- Documented requirements of the application
- Verification and validation (V&V) records to support use for the application
- User documentation
- Software configuration management plan.

These minimum requirements established a threshold to which existing RTDP software would need to be brought in order to qualify for use in LDUA.

In addition to the minimum requirements, WHC QA requires the following of software developed specifically for an application:

- Verification of the requirements
- Documentation and verification of the design
- Documentation and verification of the design implementation.

These additional requirements would have to be met for any new software created for the LDUA.

The WHC QA program uses a graduated scale of impact levels in determining the stringency of the QA provisions applied to an activity and establishing the signature authority required for approvals. There are four levels in the WHC scale. Level 4 is the lowest and is applied to activities that are not quality affecting, level 3 is applied to activities that can adversely affect equipment or work, and levels 2 and 1 are applied to activities that can have significant adverse environmental or safety consequences.

A preliminary evaluation determined that LDUA software would be mostly impact level 3, with some level 4.

## THE APPROACH

A single document addressed all parts of the development of the LDUA software rather than creating a separate project plan, SQA plan, V&V plan, and configuration management plan. This was a more straightforward approach for everyone, especially for the software suppliers. There would be only one document to follow rather than several different plans that would require cross comparison and interpretation.

The LDUA software development plan describes the following:

- Activities
- Deliverables
- V&V provisions
- Methodologies and techniques
- Configuration management provisions
- Responsibilities
- Quality assurance provisions

The heart of the LDUA software development plan is the life cycle model, which is an orderly progression from requirements to operating software in a series of steps or phases. Most of the activities, deliverables, and V&V provisions of the software development plan were organized around the life cycle. The life cycle was synchronized to the overall LDUA project schedule, which automatically defined the schedule for software development.

## THE NEGOTIATIONS

The next step was to obtain buy-in from the other participants. An outline software development plan was prepared as a strawman and discussions were undertaken with the suppliers who would be providing mission-critical software. These discussions provided WHC an opportunity to brief the suppliers on what WHC expected of them, to assess their experience and expertise, and to improve the nascent software development plan. The discussions provided the suppliers with an opportunity to plan for their needs, funding requests, and staffing to support the SQA effort. The discussions also helped dispel the fear of the unknown for some of the suppliers who had never used formal SQA before.

There was substantial acceptance of the activities and deliverables of the software development plan, but vigorous negotiation concerning the division of responsibilities among WHC and the suppliers. The final consensus was that WHC would be responsible for the integrated system and production of the system-level deliverables, and the suppliers would each be responsible for the components they provided and production of deliverables appropriate to the component. A set of categories was established for software components that specifies what deliverables are required. These categories are described later in this document.

The system-level deliverables and activities for which WHC will be responsible include the software requirements specification, software design description, user documents, test documents, and validation testing. These will apply to the total integrated system and be produced in a format consistent with WHC standards and practices.

Component deliverables and activities for which the suppliers are responsible include technical reference documents and component testing. Technical reference documents may include an operating manual, an application programming interface (API) description, or a software maintenance manual. These will apply only to the software component, and will be in the supplier's format.

Suppliers are not required to become independently expert in NQA-1, SQA, or WHC standards and practices in order to provide an acceptable product with suitable documentation. They need only follow the software development plan.

## THE LIFE CYCLE

The life cycle for the LDUA software development plan has a Requirements Phase, a Design Phase, an Implementation Phase, a Testing Phase, and an Operations and Maintenance Phase. These phases may overlap. This section describes each of the phases according to its activities, deliverables, and V&V provisions.

The Requirements Phase establishes detailed requirements for the integrated software from the LDUA system function and requirements.

Deliverables:

- Computer software requirements specification (CSRS)
- Software test plan (outline)

Verification and validation:

- Software requirements review

The Design Phase translates the detailed requirements into a computer software design description, which is a description of the integrated software that shows its breakdown into software components and describes their function and interfaces. Some software components will be further broken down into their constituent modules.

Deliverables:

- Computer software design description (CSDD)
- SW component design description
- User documents (draft)
  - Software operation manual
  - Technical reference documents
- Software testing documents:
  - Software test plan (draft)
  - Software test procedures (outline)
  - Software test specifications (draft)

Verification and validation:

- Software design review

The Implementation Phase produces working software components from the design description by programming from scratch, using or modifying existing software, or purchasing commercial software.

Deliverables:

- User documents
- Software operation manual
- Technical reference documents
- Computer software
- Software component test suite
- Software testing documents:
  - Software test plan
  - Software test procedures
  - Software test specifications
- User training

Verification and validation:

- Software implementation review
- Software component testing
- Code evaluation (optional)

The Testing Phase integrates software components into the LDUA system in the cold test facility at the Hanford Site and validates the software by formal testing of the complete system.

Deliverables:

- Verification and validation report
- Verification and validation records

Verification and validation:

- Validation testing
- Verification and validation review

The Operations and Maintenance Phase uses software for its intended purpose. Residual errors are removed and enhancements may be added during the useful life of the software.

Deliverables:
(modifications repeat previous phases and will generate the same deliverables, as appropriate)

Verification and validation:
(modifications repeat previous phases and will require the same V&V provisions)

## CONFIGURATION MANAGEMENT

Configuration management is the process of identifying the items that make up the software product, and controlling their release and the changes made to them. Configuration management ensures that only software of known derivation is used for the application. It is fair to say that without effective configuration management, all other SQA efforts are meaningless because they could not be confidently associated with a particular piece of code.

The configuration items for the LDUA software include all the deliverables noted in the life cycle description. These include documents as well as code. Westinghouse Hanford Company has mature standards and organizations for document control that more than adequately support the LDUA project. The software development plan simply invokes them. There are, however, specific provisions in the plan for control of code. These provisions specify a system for identifying each software item and its revision level, for releasing them, and for reporting and correction errors. A software custodian is to be identified for the LDUA software who's responsibility will be for physical control of the software and its access.

A software component will be placed under WHC configuration management when it is accepted for integration into the whole system. Prior to that time, the supplier is responsible for maintaining control of the configuration items, but they must be furnished in a format compatible with the configuration management provisions specified in the software development plan.

Change control will be under the authority of the WHC cognizant engineer during integration, but the supplier will be responsible for performing any code modifications.

## SOFTWARE COMPONENT CATEGORIES

The software development plan divides software components into four categories according to how the component will be supplied. There are different deliverables and V&V activities for each category.

Standard Commercial- This category of software is available off the shelf in the open market and is fully supported by the vendor. The vendor's standard documentation is acceptable.

Special Commercial- This category of software is supplied by a commercial vendor, but it is neither an off-the-shelf product nor is not supported by the vendor. At a minimum, it must have technical reference documents. Other provisions depend on the nature of the software component and the contractual relationship with the supplier. The software development suggests a set of things as a goal, but leaves the specific decision up the cognizant engineer and project management.

Existing RTDP- This category applies to software that has been developed under the RTDP. The supplier must provide with it technical reference documents and a component test suite, however, no design or implementation verification is required.

Application Specific- This category applies to software that will be developed new by WHC or one of the other DOE contractors in the consortium. It will require documentation and verification for both design and implementation, technical reference documents for the user, and a component test suite.

## TECHNIQUES AND METHODOLOGIES

There are a host of techniques and methodologies in the software engineering field that apply to system analysis, program design, testing, and so forth. There are also many systems of software metrics that attempt to objectively measure quality, or the effect of quality-improving methods. For the most part, these techniques are valuable and can improve the quality of a product if properly applied. Unfortunately, they are sometimes complicated and arcane, difficult to master, and expensive to implement. In any case, the LDUA project could not afford to support more than a few techniques chosen on the basis of the plan's primary quality goals.

One technique called for in the software development plan is rapid prototyping for user interfaces. This will quickly put something into the hands of end users that looks very much like the finished system. The users can experiment with it and communicate their needs and preferences effectively to the design team while the product is malleable. Rapid prototyping addresses the quality goal of usability.

Another mandated technique is code evaluation, which involves a set of procedures and error-detection techniques for reading of code by a group of people. The two types of code evaluation to be used for LDUA software are code inspections and code walkthroughs. Code evaluations have the reputation of being one of the most effective techniques for improving code, and experience at WHC supports this. Suppliers were concerned about their demand on resources, so a decision was made during the negotiation process to limit the demand to critical modules, and allow broader use at the discretion of the cognizant engineer and project management.

Coding standards are mandated by the plan to help ensure maintainability. The coding standard sets general guidelines for format and content of the code, but does not set any stylistic requirements. For example, it will require that descriptive headers be provided for files and functions, but will not specify the layout of those headers.

There was a strong desire by WHC to mandate the use of structured methodologies for analysis and design of the software, however suppliers were inexperienced and the methodologies can be time-consuming to implement. It was decided, though, that there was value in doing some structured design rather than none at all, so the software development plan asks that some kind of hierarchical graphic decomposition technique be used for designing application-specific software at the overall integrated system level. It was not workable to require a particular methodology, so the choice is left up to the supplier. One supplier chose to obtain training based on WHC recommendations, which is an excellent way to achieve standardization without it being legislated.

## CONCLUSIONS

The chosen approach has resulted in a workable plan that achieves a balance between the need to achieve a pre-defined level of quality and the limited available resources, and is supported and accepted by the participants and accommodates their needs.

This approach is not fancy, and it stops short of invoking much of the body of software engineering techniques that are promulgated today as correct practice. However, the plan does ensure that the LDUA software will be under control rather than out of control. In a project with as diverse a group of players as the LDUA, and with software as developmental, that must be considered a victory.

## REFERENCES

1. ASME, 1989, *Quality Assurance Program Requirements for Nuclear Facilities*, ASME NQA-1, American Society of Mechanical Engineers, New York, NewYork.

2. ASME, 1989, *Quality Assurance Requirements of Computer Software for Nuclear Facility Applications*, NQA-2a, Part 2.7, American Society of Mechanical Engineers, New York, New York.

END

DATE
FILMED
3/7/94